REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)*
1 APR 1999

2. REPORT TYPE
FINAL

3. DATES COVERED *(From - To)*

4. TITLE AND SUBTITLE
EXTENDING THE SYNTHETIC ENVIRONMENT DATA REPRESENTATION AND INTERCHANGE SPECIFICATION (SEDRIS) FOR THE REPRESENTATION OF SENSORS IN THE SYNTHETIC ENVIRONMENT

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

6. AUTHOR(S)
CPT PATRICK E. CONNORS II

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
US ARMY STUDENT DETACHMENT
FT JACKSON, SC 29207

10. SPONSOR/MONITOR'S ACRONYM(S)
USASD

11. SPONSOR/MONITOR'S REPORT

12. DISTRIBUTION/AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE

19990427 120

13. SUPPLEMENTARY NOTES
Thesis for the partial fulfillment of the requirements for the degree of Master of Science from the Department of Industrial Engineering and Management Systems in the College of Engineering at the Univerity of Central Florida, Orlando, FL

14. ABSTRACT
The Synthetic Environment Data Representation and Interchange Specification (SEDRIS) attempts to account for the data types and their relationships that are used to describe the synthetic environment in DOD. The goal of SEDRIS is the loss-less and unambiguous transfer of data from one database to another across the full range of M&S applications, including sensors. This research explores the sufficiency of the SEDRIS Data Model in terms of the loss-less, unambiguous transfer of sensor simulation data. The primary output of this research was to extend the capability of the current SEDRIS Data Model to more fully provide for sensor representation based on primitive environmental factors that could be generalized to other domains. This research investigates the current SEDRIS Data Model's capabilities concerning the representation of sensor-related properties, examines a representative sample of popular sensor models, documents the development of a mapping and analysis process, and describes the implementation, demonstration and successful testing of a portion of the proposed extended SEDRIS Data Model.

15. SUBJECT TERMS
SYNTHEITIC ENVIRONMENT; SEDRIS; SENSORS; INTERCHANGE EXPERIMENT; SENSOR-RELATED ENVIRONMENTAL ATTRIBUTES

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
UNCLASS	UNCLASS	UNCLASS	UL	225	CPT PATRICK E. CONNORS II
					19b. TELEPHONE NUMBER *(Include area code)* (407) 366-6772

EXTENDING THE SYNTHETIC ENVIRONMENT DATA REPRESENTATION AND INTERCHANGE SPECIFICATION (SEDRIS) FOR THE REPRESENTATION OF SENSORS IN THE SYNTHETIC ENVIRONMENT

by

PATRICK EDWARD CONNORS II
B.S.B.A. John Carroll University, 1989

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science
in the Department of Industrial Engineering and Management Systems
in the College of Engineering
at the University of Central Florida
Orlando, Florida

Spring Term
1999

ABSTRACT

The Department of Defense (DOD) leads the world in the study of simulation interoperability. The evidence of this leadership is the creation of Distributed Interactive Simulation (DIS) and High Level Architecture (HLA) interoperability standards. Their experience has indicated that a critical *precondition* to achieving successful interoperability is to start with a correlated initial environment. Data interchange, therefore, is a central element for achieving interoperability between distributed, heterogeneous training system networks. In recognition of this need and the desire to achieve efficiencies throughout DOD, the Defense Modeling and Simulation Office conducted research to develop a research data model described as the Synthetic Environment Data Representation and Interchange Specification (SEDRIS). SEDRIS attempts to account for those data types and their relationships that are used to describe the synthetic environment in the DOD. The goal of SEDRIS is the loss-less and unambiguous transfer of data from one database to another across the *full* range of M&S applications, including sensors.

This research explores the sufficiency of the SEDRIS Data Model in terms of the loss-less, unambiguous transfer of sensor simulation data. This research explores issues stemming from the fundamental question: Does the SEDRIS Data Model adequately provide for sensor representation? This research indicates that a gap exists between the

current SEDRIS (1.04d version) research model and what is needed based on the more general sensor requirements derived from primitive environmental factors. Essentially, the current SEDRIS Data Model does not sufficiently provide for sensor representation.

The primary output of this research was to extend the *capability* of the current SEDRIS Data Model to more fully provide for sensor representation based on primitive environmental factors that could be generalized to other domains. This research investigates the current SEDRIS Data Model's capabilities concerning the representation of sensor-related properties. The research examines a representative sample of popular sensor models and tools with respect to a categorization based on the electro-magnetic spectrum. The research developed a mapping and analysis process that not only resulted in recommendations to extend the SEDRIS Data Model to provide more fully for sensor representation but also provides a proven methodology to extend the model in the future. Finally, this research implemented, demonstrated and successfully tested a portion of the proposed extended SEDRIS Data Model.

This research will immediately benefit members of the sensor simulation community who need to use SEDRIS as an interchange format. Further, it provides extensions to the general environmental model that encompasses a wide spectrum of sensor systems based on primitive environmental elements. Additionally, this research develops a step-by-step process that can be generalized and applied to areas other than the sensor community. By documenting a proven, effective process, it is also intended to serve as a template that newcomers to the SEDRIS community can follow to reduce the time it takes to understand and extend SEDRIS to suit their individual needs.

ACKNOWLEDGMENTS

This thesis effort would not have been possible without the assistance of many outstanding simulation professionals. I would like to sincerely thank Dr. Mike Proctor. As my advisor, editor and instructor, he was instrumental in my success. His proven method of coordinating a student's thesis work with an industry partner's current simulation project leads to a positive experience for everyone involved. His ambitious goal-setting and extreme willingness to give of himself kept me on track throughout this two-year effort.

As the sponsor for this research, Dr. Wes Braudaway of the Science Applications International Corporation (SAIC) demonstrated a true concern for building and maintaining successful relationships between academia and industry. The research team at SAIC – Orlando welcomed me as one of their own and each member of the team went above and beyond the call of duty to support my efforts by providing their guidance, experience, and mentorship. Additionally, SAIC's willingness to provide the necessary resources for this research is a shining example of a fine corporation with a true commitment to improving the future of the modeling and simulation community.

The SEDRIS Team is an absolutely amazing group of talented individuals from both government and industry. These men and women, who are normally fierce competitors for market share during business hours, forged a unique working relationship

that continues to benefit the entire simulation industry. I am confident that their collective contributions will be highly regarded for years to come. I would like to single out Dr. Paul Berner, Dr. Paul Birkel, and Farid Mamaghani for contributing to my success more than they will ever know by providing their simulation expertise through written documents, oral presentations, and personal assistance.

Two SEDRIS Team members clearly stand out above the rest. Bill Horan, a simulation engineer at SAIC, and Long Nguyen, a sensor expert from the Naval Air Warfare Center – Training Systems Division, provided critical assistance to this work. I cannot thank them enough for their extreme dedication, selfless service, and concern for my success. They are true subject matter experts – knowledgeable professionals in their respective fields, gifted teachers with the ability to explain complex subjects, and caring individuals that ask nothing in return.

Finally, my wife, Laura, provided me with a level of unwavering support during this scholastic journey that allowed me to focus on my research efforts. She was my constant source of inspiration and motivation. Once again, it was a true team effort.

To all of you, I owe a debt of gratitude.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

AFASC	Air Force Aeronautical Systems Center
AGR	Air-To-Ground Ranging
API	Application Program Interface
ASNE	Air and Space Natural Environment
ATS	Aircrew Training System
BRDF	Bi-Directional Reflectance Function
CCTT	Close Combat Tactical Trainer
CGF	Computer Generated Forces
CMY	Cyan Magenta Yellow
DARPA	Defense Advanced Research Projects Agency
DBGS	Database Generation System
DBS	Doppler Beam Sharpening
DGIWG	Digital Geographic Information Working Group
DIGEST	Digital Geographic Information Exchange Standard
DIS	Distributed Interactive Simulation
DMA	Defense Mapping Agency
DMSO	Defense Modeling and Simulation Office
DOD	Department of Defense
EM	Electromagnetic
EO	Electro-optical
FACC	Feature and Attribute Coding Catalogue
FIR	Far Infrared Region
FLIR	Forward Looking Infrared Radar
GCC	Geocentric Coordinate System
GCS	Global Coordinate System
GDC	Geodetic Coordinate System
GEI	Geocentric Equatorial Inertial Coordinate System
GSE	Geocentric Solar Ecliptic Coordinate System
GSM	Geocentric Solar Magnetic Coordinate System

GTDB	Generic Transformed Data Base
HF	High Frequency
HLA	High Level Architecture
HSV	Hue Saturation Value
IR	Infrared
LCC	Lambert Conformal Conic Projected Coordinate System
LF	Low Frequency
LMTDS	Lockheed Martin Tactical Defense Systems
LSR	Local Space Rectangular Coordinate System
LWIR	Long Wavelength Infrared
M&S	Modeling and Simulation
MF	Medium Frequency
MMW	Millimeter Wave
MSEA	Modeling and Simulation Executive Agent
MSMP	Modeling and Simulation Master Plan
MWIR	Mid-Wavelength Infrared
NAWC-TSD	Naval Air Warfare Center – Training Systems Division
NIMA	National Imagery and Mapping Agency
NVD	Night Vision Device
NVESD	Night Vision and Electronic Sensors Directorate
NVG	Night Vision Goggles
OEA	Ocean Executive Agency
OMT	Object Model Technique
OTW	Out the Window
PRA	Photon Research Associates, Inc.
PS	Polar Stereographic Projected Coordinate System
PTN	Paint-the-Night Infrared Sensor Simulation
RBGM	Real Beam Ground Map
RCS	Radar Cross Section
RGB	Red Green Blue
RP	Red Phosphorus
RSF	Radar Significant Factor
SAC	SEDRIS Attribute Code
SAF	Semi-Automated Forces
SAIC	Science Applications International Corporation

SAM	SEDRIS Associates Meeting
SAR	Synthetic Aperture Radar
SCC	SEDRIS Classification Code
SCR	SEDRIS Change Request
SDCS	SEDRIS Data Coding Standard
SE	Synthetic Environment
SEDRIS	Synthetic Environment Data Representation and Interchange Specification
SIF	SSDB Interchange Format
SIMNET	Simulation Networking
SM	Solar Magnetic Coordinate System
SME	Subject Matter Expert
SOF	Special Operations Forces
SSC	SEDRIS State Code
SSDB	Standard Simulator Database
STF	SEDRIS Transmittal Format
STM	Sensor Texture Maps
STRICOM	Simulation, Training and Instrumentation Command
SWIR	Short Wavelength Infrared
TEP	Terrain Execution Plan
TFR	Terrain Following Radar
TM	Transverse Mercator Projected Coordinate System
TMPO	Terrain Modeling Project Office
TTS	Tank Thermal Sight
TTS	Thompson Training & Simulation
UHF	Ultra High Frequency
USD(A&T)	Under Secretary of Defense for Acquisition & Technology
UTM	Universal Transverse Mercator
UV	Ultra-violet
VHF	Very High Frequency
WP	White Phosphorus

CHAPTER 1

THE SYNTHETIC ENVIRONMENT AND SENSORS

1.1 Synthetic Environment Domain

All too often in rapid growth industries, certain words or phrases fall prey to overuse. This situation usually leads to ambiguity. In the training simulation industry, the term *synthetic environment* is quickly becoming the next victim of overuse. The expression synthetic environment conjures up different ideas to different people working in the simulation community. It is now necessary to start each discussion concerning a synthetic environment with a context and definition for the particular topic at hand. So the question arises – what does the synthetic environment include for this thesis? It is helpful to start with the synthetic environment domain.

> In the broadest sense, a domain is defined as a sphere of action, thought or influence. Within the simulation/ … /computer science communities, a domain is considered to be a bounded area of application – therefore, a synthetic environment domain is a representation of the natural environment of entities, actions, and interactions comprising the set of interrelated processes used by individuals and organizations to accomplish mission tasks.
>
> The synthetic environment domain exists within another domain: today's training systems domain. The training systems domain encompasses a variety of networked, heterogeneous computer-based systems intended to accomplish a common training mission. These training systems … may accommodate training by a single trainee, a team of trainees, or a collection of teams…. Bounding the entire training system domain is the synthetic environment component that provides the playing field for a training exercise. (U.S. Army Simulation, Training, and Instrumentation Command [STRICOM], 1998a, p. 2)

1.2 Synthetic Environment Definition

What constitutes a synthetic environment (SE)? According to James (1997b), today's definition of SE constitutes more than just the visual scene representation it once meant. As used in current training systems architectures, SE means the total simulated representation of the battlespace created to support the training exercise. Mamaghani (1998a) defines a SE as

> an integrated set of data elements, each describing some aspect of the same geographical region. It often includes additional data describing simulation elements and events expected to take place during the interactions in that environment. For example, data representing trees in a forested region may be found in a database; but in addition, the geometry of vehicles that might drive through the trees during a simulation would also be found in the SE. (p. 101)

Two categories of objects are represented in the SE: the natural environment with its cultural and terrain features, and the warfighting entities present in a battlespace (James, 1997b, p. 1). The Defense Modeling and Simulation Office (DMSO) Modeling and Simulation Master Plan (MSMP) states that the SE covers all environmental domains – terrain, ocean, atmosphere, and space. The SE "allows the visualization of and immersion into the environment being simulated" (Department of Defense [DoD], 1995, Ch. 4) for the training exercise.

> The [Modeling and Simulation] Master Plan further defines the environmental representation as an authoritative representation of all or a part of the natural environment including permanent or semi-permanent man-made (i.e., cultural) features.... The objective of the creation of a training system is to immerse the warfighter in a synthetic environment that accurately simulates the anticipated terrain, environmental conditions and threat. (James, 1997b, p. 2)

For this thesis, the definition of synthetic environment is the simulated natural environment usually representing a true geographic location in the world including the

tactically significant natural and cultural features along with the external, physical representations of the warfighting entities (STRICOM, 1998a, p. 5).

In addition to the visual aspects of the natural environment and objects on the battlefield, the SE now contains non-visual information to allow entities under computer control to properly interpret and navigate the environment. Mobility data, for example, is non-visual information captured in the SE terrain that tells an entity where and how well it can maneuver. It does this by identifying the terrain type (e.g., asphalt road, muddy soil, dense forest, swamp) that is currently being encountered. These non-visual attributes are the key ingredient for SE improvements – going from a realistic visual environment to a SE that is visually realistic *and* non-visually robust in terms of information about the environment.

1.3 Synthetic Environment Database Evolution

The improvements made to the synthetic environments are due to the enhanced capabilities in today's training systems. With the improved computational capabilities of computer systems, today's simulation training systems demand more information from the SE to operate effectively. For many years, this was not the case. The training audience thought that to have a reasonably realistic "out-the-window" view from their vehicle simulator was a major enhancement. Then, the only sensors that needed information were the trainee's eyes because there was very limited interaction between any simulated sensors and the simulated environment (Welch, 1997b, p. 2). In addition, "the level of detail or fidelity of the environmental scene was not required to be great

since most training systems using visual scene generators were high-altitude, fast moving aircraft trainers" (James, 1997b, p. 2).

A major impact on the SE database evolution was the development of Computer Generated Forces (CGF) and Semi-Automated Forces (SAF). According to STRICOM (1998a), CGF entities are totally under the control of their dynamic and reasoning software models. The CGF entities react automatically to the trainees' actions and the environment without input from the training system's instructors or operators. SAF entities also use dynamic and reasoning models, but they take a limited amount of high-level input from the operators. As an example, a SAF operator may issue a command to a simulated tank platoon entity to "move down road X and take a defensive position at location Y." This simulated entity cannot "see" that the chain of polygons that are gray or black with a yellow stripe down the middle that head in a southerly direction is a road. The model for the SAF or CGF entity must use information from the SE's database that allow it to identify the surface as a "road" as well as the road's direction, width, weight capacity, surface material type, etc. In addition to the linear feature that is visually represented as a road, the topological data and its attributes that define the road and terrain need to be included in the database so that the CGF models can use this information directly without having to derive its existence (p. 7).

The next logical step was to make simulations useful for sensors other than the eye. Sensor simulations impacted the SE database by requiring data and attributes that reflect the environment's state as perceived by the sensor's models. But the attributes of

the existing data types of the SE used for visual scene generation were not adequate to simulate sensors. In some cases, entirely new data types were needed.

> The sensor simulation models needed information that differs from the data that represents the natural environment. For instance:
> - Thermal sensors can be simulated by presenting displays to the trainees with different color attributes than used for normal views
> - Sonar hydrophone sensors need sound pressure arrival angle information determined by the ocean model
> - Simulated light intensification sensors need environmental data representative of their "view" of the real world
> - Electromagnetic sensors require propagation and location information from sources that may be beyond visual range. (STRICOM, 1998a, p. 6)

This brief evolutionary description of the SE database highlights the importance of the relationship between the surface geometry and the topological data represented in the environment. These relationships are critical in ensuring that the run-time databases, which are derived from the SE databases, will be correlated so that all *views* of the environment are the same. This *view* is obviously most important to the CGF and sensors that do not *see* the battlefield, but must use the data representations and imbedded relationships to correctly interpret the environmental conditions to determine their action or output (STRICOM, 1998a; Mamaghani, 1998a). James (1997b) provides an excellent example of why the database must have a single, coherent representation of the SE.

> The data object called a "tree" that is represented as polygon data types for the image generator must correlate with the representation as a topological point feature for the CGF models. The tree that a trainee sees out the window of his vehicle must be the "same" tree (i.e., same location, same height, same intervisibility, etc.) that the CGF vehicle is hiding behind. The correlation issue is not only critical for the stand-alone training system, but is extremely critical when two training systems are interoperating on the same training exercise using the "same" synthetic environment. (p. 5)

One of the most important methods to enhance realism in an *interactive* simulation is including the sensor's capabilities in the simulation. In the real world,

sensors "see" based on the physical properties of objects and the environment. In an interactive simulation using a SE, this is difficult to replicate for all of the participants. How do the simulated sensors use the SE's non-visual information to interpret the environmental conditions? In other words, how does the SE "tell" the sensor what to do, how to act, or appear, when that sensor "sees," or "paints" a terrain feature or model in the environment? What are the physical characteristics of the materials or objects in the SE and what does their physical composition "tell" the sensor simulation? To begin to answer these questions, it is necessary to understand the characteristics of both real sensors and sensor simulations.

1.4 Sensors

A sensor is defined as "any of various devices designed to detect, measure, or record physical phenomena, as radiation, heat, blood pressure, etc., and to respond, as by transmitting information, initiating changes, or operating controls" (Guralnik, Webster's New World Dictionary, 1970). Although all sensors in the vast population referenced by this definition are important, those that are (or will be) used in simulations are of concern. But that is still too general.

Since the correlation issue is most critical when two simulation training systems are interoperating, the only sensors that will be considered for study are those that are part of a system or trainer used in simulations that are interoperable. The term interoperable needs some detailed explanation, which is done later in section 1.5. For now, the definition of an interoperating simulation in its purest form is one where

multiple players are concurrently participating together in a training event. These simulations can be virtual, constructive, or a combination of both types. Also included in this broad group are simulations that will share a SE database prior to runtime. This implies that even some stand-alone simulations or simulators will be considered interoperable. This study determines if simulations or simulators meet this general definition for interoperability. This determination of simulation interoperability is labeled as the "interoperable test."

In today's simulation community, most of the sensor simulations that satisfy the "interoperable test" are modeled from military-oriented sensor systems. These sensor systems can be categorized in several different ways. What is the best way to organize them to ensure thorough consideration of the population? Almost every modern weapon system or platform has some means of detecting a target. To detect a target, the system must be able to sense some unique characteristic that differentiates it or identifies it as a target.

> One such characteristic is the energy that is either emitted or reflected by the target. This energy may be in several forms, including electrical, audio, heat, or visible light. A characteristic common to all energy forms listed above is their manner of propagation. That is, they all propagate in the form of traveling waves and as such can be defined and categorized by their frequency and wavelength. It is the function of the sensor system to detect the appropriate energy form and to furnish the appropriate information thus obtained to the other components of the weapon system. (Frieden, 1985, p. 5)

So it seems most appropriate and most precise to categorize sensor systems by their characteristic frequency or wavelength. This is typically accomplished using the electromagnetic spectrum.

1.4.1 Electromagnetic Spectrum

The electromagnetic (EM) spectrum classifies all energy that moves at the constant velocity of light in a wave pattern. It is a continuous scale. For convenience, the EM spectrum is divided into a number of bands such as x-ray, infrared and microwave. These arbitrary divisions are made in part because of the different types of sensing systems used to detect the energy (Crum, 1997). Differences arise when it comes to the units of measure along the EM spectrum. In the science/engineering community, the radar sector uses frequency, measured in Hertz, while the electro-optics sector prefers using wavelength, measured in meters. Although the conversion between the two is simple, the EM spectrum is usually depicted with both measurement scales as shown in Table 1.4.1.

This study uses wavelength for ease of notation and consistency. As mentioned, the base unit of measurement is meters, but some wavelengths are very small in the EM spectrum bands. The convention is to use centimeters (cm) for 10^{-2}, millimeters (mm) for 10^{-3}, micrometers (μm) for 10^{-6}, and nanometers (nm) for 10^{-9}. A careful inspection of the EM spectrum in Table 1.4.1 shows that the bands are not uniform in size. Likewise, sensors do not operate in (or use) each band equally; some regions within the EM spectrum are used heavily for sensor operation, and although the EM spectrum is continuous, some portions are not used at all.

Table 1.4.1

The Electromagnetic Spectrum Measured in Frequency and Wavelength

Description	Frequency	Wavelength
High Frequency (HF)	3 – 30 MHz	100 – 10 m
Very High Frequency (VHF)	50 – 100 MHz	6 – 3 m
Ultra High Frequency (UHF)	400 – 1000 MHz	75 – 30 cm
Microwaves	$3 \times 10^{9} - 10^{11}$	10 cm – 3 mm
Millimeter waves	$10^{11} - 10^{12}$	3 mm – 0.3 mm
Infrared	$10^{12} - 6 \times 10^{14}$	0.3 mm – 0.7 μm
Visible light	$6 \times 10^{14} - 8 \times 10^{14}$	0.7 μm – 0.4 μm
Ultra-violet	$8 \times 10^{14} - 10^{17}$	0.4 μm – 1 nm
X-rays	$10^{17} - 10^{19}$	1 nm – 10^{-13} m
Gamma rays	> 10^{19} Hz	< 10^{-13} m

(Bhatti, 1994)

One band that is heavily used is the infrared band. As Table 1.4.1 shows, the infrared band covers wavelengths from 0.7 μm to 0.3 mm, but it is further classified into three to five smaller regions depending on the reference you happen to open (Bhatti, 1994; Crum, 1997; Dynetics, Inc., 1991; Frieden, 1985; Holst, 1995; Rogotto 1993). Four regions are used to describe the EM spectrum for this research. Figure 1.4.1 graphically illustrates the different band sizes and includes the expanded infrared bands. The short wavelength infrared (SWIR) band covers wavelengths from 1.1 to 2.5 μm; the mid-wavelength infrared (MWIR) band covers approximately 2.5 to 7.0 μm; the long wavelength infrared (LWIR) band includes the region from 7.0 to 15 μm; and the far infrared region (FIR) band is from 15.0 μm to 0.3 mm (Holst, 1995). The two specific areas in the infrared band that are most used are 3 to 5 μm in the MWIR band and 8 to 12 μm in the LWIR band.

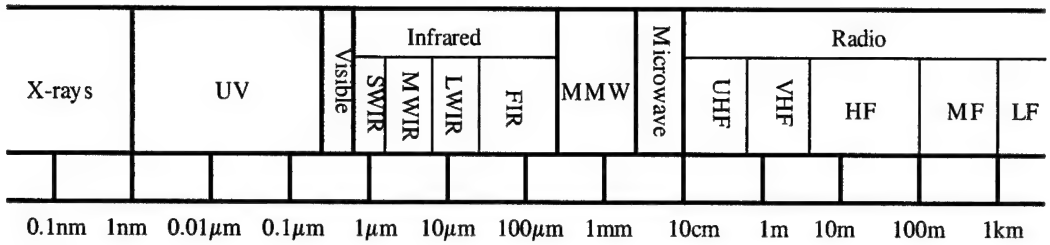


Figure 1.4.1. The Electromagnetic Spectrum

Another area of inconsistency in the science/engineering community regarding the EM spectrum is how the bands are labeled and precisely where the bands are divided. Depending on the source, you may find the bands labeled by names (as in Figure 1.4.1), letters, or even numbers. The boundaries between the bands vary based on the way they are labeled and the precision necessary for the particular field of study. For this work, however, those small differences are inconsequential; the important point to understand is that EM sensor systems can be classified by where they fall on the EM spectrum.

1.4.2 Sensor Categories

The next question to answer is where do the military-oriented sensors being used in simulations fall on the EM spectrum? The simulations used today and in the near future all seem to gravitate towards one of only a few broad categories. The "interoperable test" was applied to the military services and their simulations. The simulations and simulators that failed the test were part-task trainers, stand-alone trainers

that would never share a SE database, and test and evaluation simulations with no need for interactivity. Those that passed the "interoperable test" were:

- Virtual simulations, including manned flight simulators (fixed wing and rotary wing), manned ground vehicle simulators (tank, armored personnel carriers, etc.), manned ship and submarine simulators.
- Constructive simulations, including computer-based individual trainers (for flight, ground vehicle, and ship/submarine), command and staff trainers, analysis simulations, and test and evaluation simulations for acquisition and training development.

So the simulations/simulators that passed the interoperable test are virtual, constructive, or combinations of both. This allowed for the study of practically every military simulation involving sensors. Additionally, the sensors used by the different services (Army, Navy, and Air Force) are very similar, if not the same sensor, but installed on a different platform. Since it is not reasonable to list every specific sensor nomenclature, function and platform, a list was compiled that groups the common types together and matches them to where they operate on the EM spectrum as depicted in Figure 1.4.2.

Looking at the sensor types on the EM spectrum shows that there are several common functional groupings as well. For example, it is fairly obvious that one functional grouping is radar sensors. Another grouping is lasers. By selecting sensor systems based on their functional grouping, a representative sample from the population of military-oriented sensor systems can be attained that is diverse along the EM spectrum.

For this study the following functional groupings will be used and referred to as sensor categories:

- Optical systems
- Lasers
- Thermals
- Radars

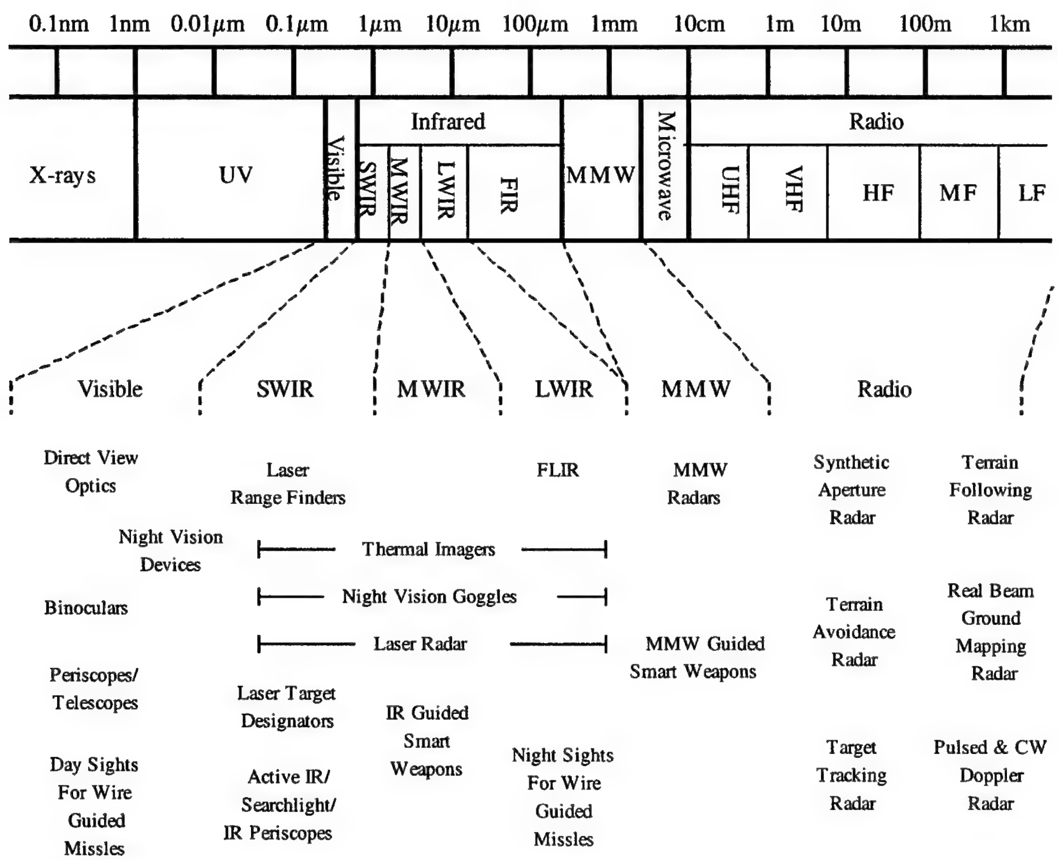


Figure 1.4.2. Sensor Types on the Electromagnetic Spectrum

(Rogatto, 1993; Dynetics, Inc., 1991)

1.4.3 How Sensor Simulations Determine Their Output

Fidelity is another term that is overused in the simulation community. It has many slightly different meanings. In fact, most source documents do not rely on just one definition. For instance, according to STRICOM (1998c), fidelity is defined as the

> "discrimination ability (level of detail)" (Birkel - Spring 96 Presentation) or "the accuracy of the representation when compared to the real world." (M&S Glossary) or "(1) The similarity, both physical and functional, between the simulation and that which it simulates. (2) A measure of the realism of a simulation. (3) The degree to which the representation within a simulation is similar to a real world object, feature, or condition in a measurable or perceivable manner." (TEP Definitions). (p. 8)

Most simulations will have a different level of fidelity from each other and from reality. Therefore it often becomes a method for distinguishing between different simulations. For this study, the measure of realism (usually called fidelity) of sensor simulations connotes something slightly different than the traditional meaning associated with fidelity. To avoid any confusion caused by using an ambiguous term like fidelity, the measure of realism for sensor simulations is defined by how the sensor simulations use the environment when determining their sensor output. There are two basic methods used in the simulation community today: those that are model-related and those that are not model-related.

The sensor simulations that are model-related have the sensors interact with the environment, then use the information gained from the environment to calculate the sensor's output. These sensor simulations have a high measure of realism because they are modeled to replicate how the corresponding real-world sensor determines it's output, i.e., by using the measurements of the physical properties present in the environment in the target area as input requirements to produce the sensor's output. To operate properly,

this type of sensor simulation needs a SE that is representationally robust. In other words, the SE must include the appropriate input requirements that the sensor can use to determine it's output.

For instance, in the real world the thermal sight on a tank measures the thermal emissions of everything (hillside, trees, targets, buildings, etc.) within its field of view. Based on these radiance measurements (from the thermal emissions), a corresponding amount of brightness (output) is displayed to the soldier. Generally, the hotter the object, the brighter it appears. In a model-related sensor simulation, the tank thermal sight uses the same basic process, but it "reads" the polygonal attribute values of emissivity and temperature, and calculates the radiance of the trees, targets, etc., in the SE. The simulation's algorithms convert these values to the gray shade colors that simulate the amount of brightness, then displays these colors to the soldier. This example illustrates that the geometries and features in the SE supporting a model-related sensor simulation must have sensor-related attributes.

Those sensor simulations that are not model-related, on the other hand, do not interact with the SE the same way. The geometries and features in the SE do not have sensor-related attributes. Instead, the most common technique is to alter the visual representation of the entire SE based on the viewing mode. For example, in addition to a polygon's primary color value, a thermal color value is also stored. When the soldier in the tank uses his thermal sight, the simulation's computer image generator switches to thermal mode and renders the SE using the stored thermal gray shade colors. Using this technique, issues such as variance due to current temperature and atmospheric effects are

not considered. So, although the degree to which the thermal display is similar to the real-world display, the general method used to derive the view is very different. These types of sensor simulations have a low measure of realism because they are not modeled upon the method used by their corresponding real-world sensors to determine their output.

A non model-related sensor simulation does not imply the simulation system is a poor product or a less effective training device. It simply means the producer had to make some tough decisions to meet the project's budget by balancing competing requirements such as processing speed, database size, and cost in dollars. But without delving into all the trade-off considerations, suffice it to say that there are legitimate reasons for the differences; and for this study sensor simulations are classified as either model-related or non model-related. In the future, this distinction may become blurred. As the computational capacity of training systems continues to increase and the cost of the processing power decreases, it is likely that almost all future sensor simulation systems hitting the market will fall in the model-related category.

1.5 The Need for Simulation Interoperability

As previously stated, interoperable simulations are important. That fact is evidenced by the decision to use interoperability as a filter to determine how to focus research attention. However, interoperability was not defined and the reasons for its importance were not explained. This section discusses what it is, and the main reasons that interoperability between simulations is both required and desired. During the

discussion in this section, simulations are used in general to investigate interoperability. Since many simulations contain sensor simulations, the same principles concerning simulation interoperability apply to sensor simulation interoperability.

"What constitutes interoperability is not fully characterized.... It is a complex and somewhat subjective issue, and its parameters are many and undefined.... In large part it is the subjective nature of interoperability that makes it difficult to translate it into technical specification" (Mamaghani, 1994, p. 109). Once again, there are numerous definitions. According to the Modeling and Simulation (M&S) Terrain Execution Plan (TEP), interoperability is "the ability of a model or simulation to provide services to, and accept services from, other models and simulations, and to use the services so exchanged to enable them to operate effectively together" (Terrain Modeling Project Office [TMPO], 1996). In the context of a military training exercise, James (1997c) says interoperability is "two training systems interoperating to present a single training exercise in the same battlespace to a geographically dispersed training audience" (p. 5).

Although both definitions describe sharing data to facilitate operating together, the first definition hints at the ability to interchange data, while the second one focuses on interactions for training purposes. Why is interoperability important? Two of the main incentives for achieving interoperability are to improve training effectiveness and reduce cost.

In the military, nothing is more frustrating and dangerous than unrealistic training. To train together in a synthetic environment, the individual players need the ability to interact with each other. Allen, Hays, and Buffardi (1986) found that higher fidelity

simulators decreased the time for learning. In other words, the more realistic the simulated training interaction, the more effective the training becomes. One critical step towards a realistic interaction is to operate in the "same" SE during the training exercise. As previously stated, the current architecture of the simulated training domain is a network of heterogeneous systems. "This architecture provides the capability for interoperability between training systems.... Achieving a high degree of correlated 'sameness' of the synthetic environment is a must to ensure that conditions for a 'fair fight' will exist between trainees in each networked trainer" (STRICOM, 1998a, p. 3). Therefore, before the exercise begins, all participants must ensure they are using a common representation of the physical environment.

The other major incentive for achieving interoperability is cost reductions. There are three ways interoperability yields savings. First, virtual and constructive simulations are less expensive than live simulation. If the training systems can work together seamlessly, and simulation training proves effective, the overall cost of training decreases. The savings are realized in terms of less fuel, repair parts, food, etc., used during field training. Second, all participants in a training exercise rarely have high fidelity equipment. So interoperation leverages existing hardware investments in legacy stand-alone simulators (Mamaghani, 1998a, p. 111). The third way cost reduction occurs is through reuse. The by-product of striving to achieve interoperability to improve training effectiveness is the opportunity to reuse SE databases. Because the SE is a critical element of training systems, they are carefully constructed, and therefore, are

expensive. Reuse saves time and money on research, development, production, and testing.

Interoperability and reuse are so important that in 1995, the Under Secretary of Defense for Acquisition & Technology (USD(A&T)) designated the Defense Mapping Agency (now called the National Imagery and Mapping Agency [NIMA]) as the DoD Modeling and Simulation Executive Agent (MSEA) for authoritative representation of the terrain environment.

> DoD Directive 5000.59 defines an MSEA as a DoD Component to whom the USD(A&T) assigns management responsibility and delegates authority for the development and maintenance of a specific area of M&S application, including relevant standards and databases, used by or common to many models and simulations. In fulfilling its responsibilities the MSEA must:
>
> a. Foster interoperability and reuse as critical elements in establishing cost-effective capabilities to build and use simulated and/or synthetic environments.
> b. Facilitate the establishment and operation of a process by which M&S developers and users have a responsive means to acquire source data from a range of data producers. (TMPO, 1996)

Additionally, one of the primary objectives of the procedures and projects outlined in the M&S Terrain Execution Plan is to ensure that the common environmental representations are reusable to the largest extent possible to promote interoperability and cost savings.

1.6 Interoperability Starts with Data Interchange

Two forms of data interchange exist in simulations: dynamic and static. According to Mamaghani (1994), dynamic interchange refers to the interchange of data during real time operations. The exchange of network packets that communicate information such as location, velocity, and state changes of entities in the virtual

environment is considered a dynamic interchange. The format and method for communicating this information is defined and agreed upon by all participants (p. 13). Distributed Interactive Simulation and High Level Architecture are examples.

Static interchange of data refers to the transfer and sharing of information that is not expected to change during runtime. An example of a static data interchange is "sharing a common description of the environment (the database) that contains sufficient data to satisfy the needs of the various simulators in an efficient and lossless manner" (Mamaghani, 1994, p. 13).

For separate simulation applications to join each other on the synthetic battlefield, they must first share a common ground truth. That means that before a simulation training exercise begins, a static data interchange needs to take place. Welsh (1997a) explains a common misperception very well.

> To share implies the two-way interchange of information and perception. No problem – I will give you my environmental data and you will give me yours. But, interchange is more than the physical swapping of data. To use the data, we must both speak the same language – so that we can truly share information. It is the common *interpretation* or perception of the environmental data, resulting from the use of a common language that defines a successful data interchange. (p. 2)

According to Birkel (1998), "Correlated initial environment is *the* critical precondition to achieving synthetic environment interoperability." The word precondition is very important. The interchange of pertinent data is often confused with interoperability. Interchange of data does not guarantee interoperability. As Mamaghani (1994) says, "Just because two systems can babble on the network and exchange data does not mean they have interoperated" (p. 13). However, the interchange process is the necessary step that enables us to address interoperability issues.

Data interchange, therefore, is a central element for achieving interoperability between distributed, heterogeneous training system networks. To successfully interchange environmental data, the interchange method must account for all data types and their relationships used to describe the SE (STRICOM, 1998a, p. 3). According to Welsh and James (1997), "the goal is a *loss-less*, *unambiguous* transfer of data from one database directly into another. This interchange capability will save dollars through synthetic environment data reuse and will improve training effectiveness through interoperability of training systems. An unambiguous, loss-less data interchange will minimize the potential inter-training system correlation concern" (p. 2).

Although data interchange for simulations in general is discussed here, the same basic rationale applies to the subset of sensor simulations. However, the goal of loss-less, unambiguous transfer of data from one database into another is even more critical for sensor simulations. Sensor simulations, like CGF, are different than visual-oriented simulations because they need information from the SE before they can compute their output. In other words, the SE has more impact on sensor simulations than other simulation models with respect to realism. If the data interchange falls short because some information was altered or there was some loss of information, the sensor will either provide incorrect output or no output at all, respectively, thereby rendering the simulation inoperable. Realizing the criticality of data interchange, chapter 2 examines interchange characteristics and methods.

CHAPTER 2

DATA INTERCHANGE AND THE SEDRIS PROJECT

2.1 Data Interchange Characteristics

To understand data interchange, it is necessary to realize the factors involved in data representation. The type of data found in the SE covers a wide range. The data is in a variety of forms describing the terrain surface itself, the complex features placed on the terrain, the dynamic objects with special 3-D model attributes and characteristics, the sensor attributes and characteristics, the atmospheric and oceanographic features and much more. Mamaghani (1998a) states that "it is the integration, infusion, and tailoring of varied data sources that creates a full SE, and sets it apart from databases that only use an existing raw data source as-is" (p. 101).

Foley, Mamaghani, and Birkel (1998) determined that there are three important factors that affect representation of the various data types that comprise the SE database from the viewpoint of the M&S application:

- Representational Polymorphism: Applications, despite sharing a common geographical area requirement, often only need a subset of the entire data set.... This means that these applications find the remaining representations of the data irrelevant to their needs and would prefer not to be burdened by these during data access.
- Representational Completeness: The method by which the same exact data type is viewed can be radically different across applications. The type itself is not the issue, but the form in which it is expected becomes important (e.g., a 3-D CAD

model, a 2-D building, or a 1-D symbol) and if not found can lead to serious divergence among … synthetic environment databases.

- Representational Efficiency: The format of a representation is often just as critical as the capabilities of the methods used for its access and the efficiency of those methods. While the individual data elements of the representation may be complete, they may not be concise or their relationships may not be explicitly maintained.

The combination of these three factors results in a potential multiplicity of data representations and implementations, all of which are important to some consuming application. The problem is that information (data elements, relationships, encoding) viewed by one user could be interpreted as noise by another. (p. 5)

2.2 Data Interchange Methods

Interchange involves consensual sharing of data as well as acceptance of a common definition of the interchanged information. "It requires that 'you get back what you put in.' The interchange medium must not attempt to add value to the data it accepts; nor, can it minimize the data so that it is less than the original submittal" (STRICOM, 1998a, p. 25). An agreement must be made to either convert one format to another or adopt a standardized transmittal medium that can serve as the basis for the interchange. The conversion concept has been tried in a variety of ways and has never achieved wide success (Foley et al., 1998, p. 5). To support the implementation of the latter concept, an effort must be made to "capture the practical data types and their relationships, and to generate the basis for a robust and efficient synthetic environment interchange mechanism that meets the stringent needs of virtual and constructive simulation systems" (Welsh, 1997a, p. 2). Without an effective interchange mechanism, most SE database interchange continues to be accomplished by the point-to-point conversion method.

2.2.1 Point to Point Conversion

Like the title suggests, this method involves the point-to-point unique conversion between two applications. If a training exercise involves six networked simulators, the point-to-point interchange method could involve up to 30 unique conversions. Figure 2.2.1 shows the duplicative paths of a point-to-point approach to data sharing and interchange.

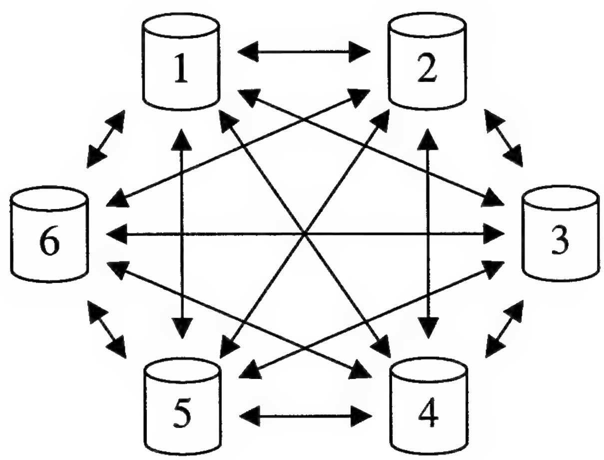


Figure 2.2.1. Point-to-Point Interchange (each arrowhead is a unique conversion) (Foley et al., 1998, p. 2)

Conversion of one system's data to another format is based upon rigidly defined database formats for both the source and target databases. The databases depicted in Figure 2.2.1 represent either producer or consumer databases or those stored in a centralized resource repository.

Welsh and James (1997) explain that because of the differing proprietary database formats, each conversion requires the development of custom data conversion software.

These point-to-point solutions are expensive, time consuming and often unreliable (p. 2). Compounding things even more, to meet the specific implementation of the target system, the converted data sets usually have to undergo additional conversions before a useable runtime database is obtained. Conversions add the risk of data loss or fidelity diminution. As the number of additional data sources increases, the number of unique conversions increases geometrically (Welsh, 1997; Mamaghani, 1998b).

At first glance, it may seem that although this method is expensive, at least the cost is a one-time expenditure for the development of the conversion software modules. Then logically speaking, if the simulators work together in subsequent training events, a cost savings could be realized. This is true in some cases, but there is more to the picture. For this to be effective, additional funding is required in two more areas. First for salaries to pay personnel to perform database maintenance and post-runtime archival, and second for overhead to pay for necessary storage facilities needed. With this in mind, another approach was started.

2.2.2 Project 2851

Recognizing the need to minimize the costs of formatting specific geographic data for inclusion in environmental databases, Project 2851 was started. It was an attempt to establish a means of interchanging the environmental data between different training system's databases (Welsh, 1997a, p. 4). This project was a tri-service effort managed by the Air Force Aeronautical Systems Center (AFASC). The AFASC served as Executive Agent for the three services. Project 2851 was started in March 1987, as a research and

development effort to design and implement standard databases and software to support DoD training simulators (Ada Information Clearinghouse [AdaIC], 1996).

The results of Project 2851 were MIL-STD-1820, Generic Transformed Data Base Design Standard, and MIL-STD-1821, Standard Simulator Data Base (SSDB) Interchange Format (SIF) Design Standard, and the creation of the Standard Simulator Database Facility at Kirtland Air Force Base in New Mexico (James, 1997c; Foley et al., 1998). The SSDB was established as the central repository for the simulator databases for the DoD training simulation community, and data was inserted and retrieved through its standard mechanism, the SIF (Welsh, 1997a).

With these two standards and their associated standard data formats, the representational data for a given geographical location could be shared between two different training systems even though the systems had different image generators (James, 1997c, p.3). For the given geographical location, the data is extracted from the SSDB and a Generic Transformed Data Base (GTBD) is produced. This GTDB is optimized for use on a specific image generator to support a particular weapon system training requirement. GTDBs can also be produced for image generators that simulate visual, radar, infrared or night vision goggle scenes. The GTDBs that are furnished to the requester will be closely correlated, providing consistency between the various sensor displays for a common geographical area (AdaIC, 1996). This data sharing would greatly reduce the costs of developing environmental databases.

According to Foley et al. (1998), the intent of the program was to allow reuse of previously generated databases, reduce the amount of data transformation required to

support various existing simulator database designs, and provide better response to user community requirements (p. 2). Project 2851 was focused on supporting the interchange of environmental data for a specific subset of environmental data types, principally the data for the generation of a visualization of the terrain and its features. The portion of the simulation community that SIF supports is the traditional virtual simulation community, specifically the cockpit-type of simulator with an out-the-window visual scene along with sensors such as radar. SIF concentrated on a data format, rather than a data model with standard access methods (Synthetic Environment Data Representation and Interchange Specification [SEDRIS] technical information web site, 1998, FAQ section).

In terms of its original objectives, Project 2851 was successful. However, in the late 1980s and the early 1990s, projects such as the Simulation Networking (SIMNET) Program and the U.S. Army Close Combat Tactical Trainer (CCTT) highlighted the design deficiencies of SIF when used by networked simulation systems. Foley et al. (1998) wrote that the most notable deficiencies were the lack of a comprehensive data model and data dictionary, software access libraries, and a focus on data interchange consumers in stand-alone visual systems and sensor simulation domains.

> Since Project 2851 was not required to meet the needs of networked simulation and new requirements have evolved since its design, it does not meet the needs of [sensor simulations] and computer-generated force (CGF) simulations, and is not well suited for high-density terrain databases. It also was not designed to support ocean, atmosphere, space, and some terrain objects described in current joint and/or networked system requirement documents. Furthermore, attempts to modify and improve SIF have been program specific, marginally successful, and generally not extendable. (Foley et al., 1998, p. 2)

Based on these factors, a new solution to support the efficient interchange of simulated environment databases was required.

2.2.3 Interchange Based on a Data Model

As demonstrated by Project 2851's deficiencies, a format-focused approach to data interchange led to ambiguous data since the underlying meaning and relationships of the data cannot be captured with just a description of the data's format. A data model, however, provides not only a clear description of the data but also defines the relationships between the data – relationships that are critical to ensuring a correct interpretation by the users (James, 1997a, p. 1).

The definition of a data model according to STRICOM (1998c) is "description of the logical relationships between data elements. Each major data element with important or explicit relationships is captured to show its logical relationship to other data elements" (p. 7). Although that technically defines a data model, James (1997a) explains the data model concept in a more understandable manner.

> A data model is a graphics-based design tool used to provide an identification of the types of data, with their attributes and relationships, that are used within a system. The use of a data model is a software engineering technique for unambiguously articulating information about the data. A data model provides a capability to articulate what kinds of things (i.e., data) are contained in the system. A data model is not an implementation of a database; rather, it is a depiction of the types of data within the system as well as a justification of why the data is needed. A data model is supported by a data dictionary which provides additional textual (i.e., non-graphical) information describing each type of data including it's attributes and relationships. (p. 1)

It is easier to communicate with a data model serving as a meta-model of the data rather than with the actual data. According to Foley et al. (1998), this is so because the meta-model describes the data through its attributes rather than through its storage formats. The data model removes ambiguity by ensuring that all types of environmental data are captured and relationships between alternate representations (e.g., feature vs.

geometry) are defined (p. 6). A data model uses a specific notation to provide a shorthand method to depict the information. This notation allows the data model to be presented in a clear, graphical manner while still providing a means to fully capture the design of the data. "The data model is a representational scheme not a format – although a data model can be easily mapped to a format. It is far more difficult to map a format to a data model" (James, 1997a, p. 2). Since a data model is a notational depiction, it can easily be examined and improved to ensure that it is a complete representation of the system's data.

The data model identifies classes of data as well as the primitive data types. The data classes depict the organizational structure of the data. The primitive data types define the fundamental data elements that are used to define the other data classes. The identification of data classes also provides the capability to describe the relationships between the classes. "All of this information – classes, primitives, and relationships – can be thought of as a specialized language with its own grammar. Given that we have a language, we can now minimize any differences of interpretation of the data contained in the data model" (James, 1997a, p. 2).

A data model also supports development of Application Program Interfaces (APIs) to be used for accessing the data. Using the APIs, producers and consumers can readily develop software tools to convert their native data to and from a data model. Therefore, a data model enables *reuse* and *interchange* of synthetic environmental data (STRICOM, 1998b, p. 9). Using the data model approach, the

Synthetic Environment Data Representation and Interchange Specification (SEDRIS) project was initiated.

2.3 The SEDRIS Project

SEDRIS is the new solution that supports the efficient interchange of simulated environment databases. The SEDRIS project was conceived and implemented to capture and provide a complete data model of the physical environment, access methods to that data model, and an associated interchange format. These SEDRIS developed mechanisms facilitate interoperability among heterogeneous simulations by providing complete and unambiguous interchange of environmental data (Foley et al., 1998, p. 1). According to STRICOM (1998b), the SEDRIS Data Model supports the full range of simulation applications (e.g., CGF, manned, visual and sensor systems) across all environmental domains – terrain, ocean, atmosphere, and space (p. 2).

According to STRICOM (1998a), "SEDRIS' primary goal is to facilitate transmission of synthetic environment data between heterogeneous training simulations to achieve a correlated environment that supports a joint training capability" (p. 4). To accomplish this goal, SEDRIS focuses on the data representation of what was used earlier as the definition for the SE, the natural environment and the warfighting models. "SEDRIS provides, through its Data Model, a standardized representation of the total synthetic environment which all providers and consumers can utilize as an intermediary between their own proprietary formats" (STRICOM, 1998a, p. 4).

2.3.1 The SEDRIS Project Objectives

The SEDRIS project is sponsored by DMSO. The M&S Master Plan, authored by DMSO, states that providing timely and authoritative representation of the environment is a core requirement in order to achieve interoperability among aggregated heterogeneous simulation systems. Toward meeting this need, the SEDRIS Project was given the general objective of solving the environmental data interchange problem. However, in accomplishing that objective, SEDRIS also solves the related interchange and reuse problems encountered by database producers and operational users (Foley et al., 1998).

In the context of promoting environmental data reuse and interoperability, the specific objectives of the SEDRIS development effort are to:

- Capture the complete set of environmental data elements and their relationships in a data model.
- Provide a software library implementing a standard API for access to data elements.
- Minimize cost to access and reuse environmental data by lowering the software barrier to entry.
- Provide a standard data interchange mechanism between database builders and consumers.
- Facilitate interoperability of networked heterogeneous simulations.
- Support reuse of environmental databases between disparate simulations.
- Use the same data model for both completed database interchange and as an access mechanism to import and export source data into and out of various database generation systems.
- Promote a common understanding of the diverse requirements and implementation choices used within the broad M&S community through education.

(Foley et al., 1998, p. 3)

2.3.2 SEDRIS Components

The full SEDRIS implementation is comprised of a set of integrated components. The Data Model is the central core to SEDRIS and allows the unambiguous description of a synthetic environment. Through the use of representational polymorphism, the model implements the capability to provide environmental data in whatever context a consumer requires. The Data Model provides for expansion to incorporate any future needs of the environmental representational community (STRICOM, 1998b, p. 3).

An API is provided with the Data Model. The API read and write routines provide SEDRIS producers and consumers a standard means for the transfer of data between their native databases and the memory-resident SEDRIS transmittal. The API interchange mechanism provides a coherent and complete interface to the data in a SEDRIS transmittal, which is structured in accordance with the Data Model. The API minimizes the software investment by data producers and consumers since they do not need to develop data and media access code and associated libraries, which are common across all applications and platforms. The API makes the underlying format transparent (but not inaccessible) to users by reflecting the view of the Data Model, as opposed to a specific media format. Use of the API ensures a loss-less interchange of data (SEDRIS technical information web site, 1998, Documentation section).

The SEDRIS interchange specification is designed to serve as a standard intermediary between differing representations and formats thereby providing a conduit for interchange. Figure 2.3.2 shows how this concept is implemented, using the same six

networked simulations as in Figure 2.2.1, through the use of a standard interface and common data model.

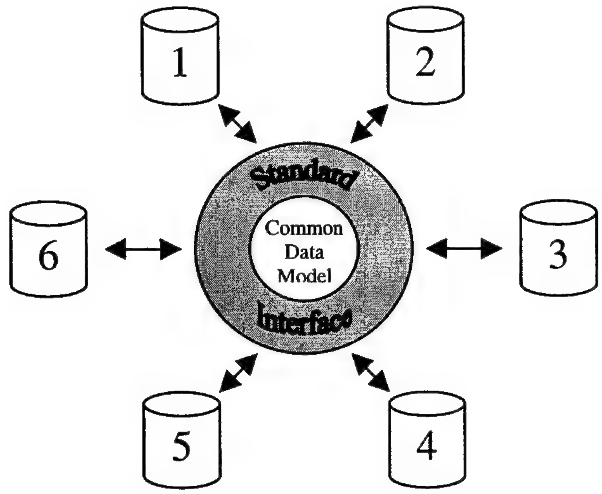


Figure 2.3.2. Interchange through SEDRIS

(Foley et al., 1998, p. 8)

Additionally, SEDRIS associate contractors have developed a set of reusable tools using the Data Model and APIs to interchange their databases using SEDRIS. These tools and applications include, among others things, common conversion utilities, data content and data syntax verification tools, helper applications, and data view and examination applications (SEDRIS technical information web site, 1998, Documentation section). Detailed information about each of the available tools and utilities can be found in the Products & Documents section under "Tools and Utilities" at the SEDRIS technical information web site.

2.3.3 SEDRIS Data Model Notation

The SEDRIS Data Model is based on Rumbaugh's Object Model Technique (OMT) notation (STRICOM, 1998b, p. 12). The SEDRIS Data Model uses some minor extensions to the OMT (i.e., extra notations) but, for the most part, the notation is OMT. The OMT notation follows the object-oriented methodology for organizing the data. "Although development of any subsequent database system does not have to also use object-oriented techniques, use of the object-oriented methodology during the analysis and design phases is very beneficial in database applications due to its data centric paradigm" (James, 1997a, p. 2).

The concept of a class of data is of primary importance in OMT notation. According to Ellis (1994), a class is a collection of fields and methods that are common to multiple objects; a class may be considered a mold from which objects are extracted (p. 21). A class identifies the attributes of the data type and the operations that can be performed on instances (i.e., objects) of the data type. James (1997a) points out that by using abstractions of data type classes, the design of a software system can be described using the same terminology as the corresponding real-world objects and their features. In SEDRIS, the Data Model contains the data classes for the *representation* of the environment, not abstractions of the objects in the environment. For instance, the SEDRIS Data Model does not contain a data class for a "tree" or a "tank" but instead contains the data classes that are used to *represent* a "tree" or a "tank" or any other kind of object found in the SE (p. 2).

In order to properly read the SEDRIS Data Model diagrams, it is essential to understand the concept of modeling representational data types. The detailed SEDRIS Data Model notation is explained in Appendix A, where two kinds of classes and three kinds of relationships found in the SEDRIS Data Model are defined and illustrated.

2.3.4 SEDRIS Data Modeling Technique

As indicated earlier, data classes depicted in an object-oriented data model are an abstraction of real-world "things" found in the problem domain. By being able to analyze and design a software system using data with real-world terminology, improved understanding and communications can be realized. In the context of the "tree" or "tank" example above, it is important to reiterate that the problem domain of SEDRIS is the *representation* of the real-world environment. According to James (1997a), this representation is accomplished through the use of databases containing the data necessary to produce both visual and non-visual representations of SEs. One set of data classes in the SEDRIS Data Model are the geometric data types used to depict a visual scene. These classes are data types such as polygons, lines, vertices, colors, textures, etc. The Data Model also contains a set of data classes that provide an alternative representation of the information contained in the geometric data types. These classes are the features data classes that use primitive data abstractions of node, edge, and face to describe point, linear, and areal data classes (p. 7).

As an example, consider the data class of a polygon, one of the geometric data classes. The polygon data class has attributes that include at least three vertices – which

are also data classes. This relationship (a "has-a" relationship) is a rule that defines the polygon data type. If it doesn't have at least three vertices, it is not a polygon. Vertex data types have additional attributes such as location and color. Taken together, this organization of data types and their attributes is a very clear definition of what constitutes a polygon and how to represent it in a visual scene. Therefore, with the proper use and interpretation of the model's notation, an unambiguous communication about a polygon is possible (STRICOM, 1998b, p.13).

The polygon data class can contain the data used to create the visual representation of a "tree" or any other object in the synthetic environment. However, the data used to define a polygon that represents the image of a "tree" in a visual scene may not be the kind of data needed by a CGF model to "see" the "tree" and make decisions based on the presence of a "tree" in the environment. James (1997a) explains that

> to support the non-visual parts of simulation applications, the SEDRIS Data Model provides alternative data types that provide other representations of "things" in the environment. These alternative representations are implemented through the use of feature data classes. At the "tree's" location in the environment, there is a point feature data class whose instance is identified as a "tree." This point feature has certain attributes, described by its use of the primitive data classes of nodes, edges, and faces, that have been organized in a specific manner to define the representation of a "tree" feature model. This information can be directly used by the CGF models to make decisions about the "tree." Should the CGF entity go around the "tree" or is the "tree" small enough to maneuver right over it? Can the CGF entity hide behind the "tree" or can the entity climb up in the "tree"? Note that the geometric description of the "tree" in polygons would not have been as readily used by the CGF models to determine the answers to the above questions. The alternative representation relationship between the geometric data to create the visual image of a "tree" and the feature data to reason about a "tree" is captured in the SEDRIS Data Model. This relationship is used to clearly define to a user the kind of "thing" to represent in the environment. (p. 8)

To further assist the reasoning models of the non-visual components of simulation applications, the SEDRIS Data Model also contains topological information about the "things" represented in a particular synthetic environment. The topology information is provided for both geometric and feature data types. The topology information provides connectivity and adjacency information about the spatial "things" in the environment. The topological information is contained in other specific data classes within the Data Model with relationships that tie this information to the geometric and feature data classes. The topological relationships assist the reasoning models by providing explicit information instead of requiring calculations to derive the information (SEDRIS technical information web site, 1998, Data Model section).

For instance, from the standpoint of a CGF model trying to determine if there is a road near the field "they" are in, a topological structure provides the identification of what environmental object is to "their" left, right, etc. Since a CGF entity cannot "see," it must make determinations through algorithmic calculations. To determine the location of a usable road or an impassable obstacle, the CGF entity requires environmental features with explicit attributes and relationships that will run efficiently on its reasoning models. The topological relationships are pre-computed and stored in the SEDRIS Data Model to speed access and use of topology information. Using topology information, a CGF entity does not have to determine if linear terrain features connect to become a road; it only has to determine if the road is going in the correct direction and if it can handle the CGF vehicle (STRICOM, 1998b, p.17).

2.3.5 Organization of the SEDRIS Data Model

The SEDRIS Data Model is contained within a multi-panel diagram. Although the panels help to organize the presentation of the Data Model to users, they are not critical to understanding the *concept* of the model. What is critical is the organization and definition of the data contained in the model. Here, only a high-level overview of the basic structure of the Data Model is presented. Volume 3 of the SEDRIS Documentation Set, *The SEDRIS User's Guide*, provides a detailed examination of the entire data model as well as technical guidance on its use. The user's guide is available in the Products & Documents section under "Documents" at the SEDRIS technical information web site.

The top level of the SEDRIS Data Model structure is summarized in Figure 2.3.5. All SEDRIS data transmittals contain an instance of the Synthetic Environment class:

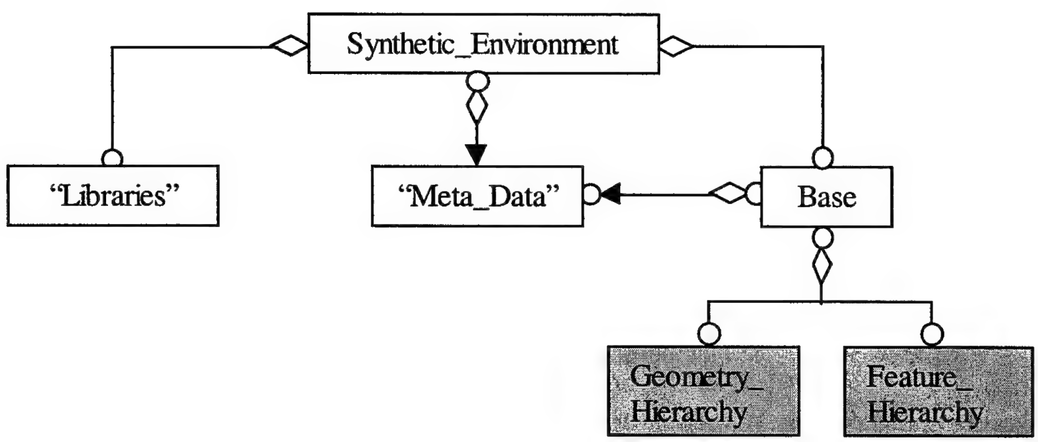


Figure 2.3.5. Top Level of the SEDRIS Data Model

(James, 1997a, p. 9)

As you can see, the Synthetic Environment class is an aggregation of other classes of data. The classes shown with their names in quotes are not actual classes from the SEDRIS Data Model, but represent a category of classes in the Data Model and are shown here for illustration purposes only. The "Libraries" Class shown is actually a set of library data classes such as Models, Colors, Symbols and Data Tables. In the Data Model, the "Meta-Data" Class shown above is in fact a set of data classes that describe the transmittal (e.g., Point of Contact at the database producer's facility, Keywords, and other identifying information).

The Base Class, which has its own descriptive set of "Meta-Data" information and is a true class from the Data Model, is the class defining the simulated world and its static features for the particular SE database. The Base is an aggregation of Geometry Hierarchy and Feature Hierarchy, which are also true classes in the Data Model. These Hierarchy classes are abstract classes (shaded) and each has an extensive set of subclasses. These subclasses define many of the remaining data types, with their attributes, that unambiguously define the makeup of the transmitted SE. These subclasses also show the relationships between geometric data and feature data, indicating when one data type can be used as an alternative representation of the other data type. In addition, topological relationships of both the geometric and feature data types are also contained within the data transmittal (James, 1997a, p. 10). The 1.04d version of the Data Model is shown at Appendix B. This 16-panel diagram indicates the explicit relationships between the all the classes in the SEDRIS Data Model.

2.4 SEDRIS Data Model Status

According to Foley et al. (1998), "the range of M&S applications addressed by the SEDRIS development effort includes training, analysis, and system acquisition and supports visual, computer generated forces, and sensor perspectives. When completed … the data interchange specification will support the pre-runtime distribution of source data, three-dimensional models, and integrated databases that describe the physical environment for both simulation and operational use" (p. 1). That is certainly the *end goal* of the SEDRIS Project. This study/development effort is part of the work that will make that goal a reality. At the start of this research in January 1998, the Data Model was immature with respect to support for sensor perspectives.

The Data Model, robust in most other areas of environmental representation, did not fully incorporate representation of the environment with respect to sensor simulation systems. The Data Model only allowed the representation of parts of non model-related sensor simulations. As an example, consider the case of representing night vision goggles (NVG). Although the Data Model included a method for representing NVG attributes, the manner in which it was accomplished was only useful to man-in-the-loop operators that could see the visual display. The computer's image generator created the visual scene by changing the color of the user's display from red-green-blue (RGB) to black and green when the NVGs were in use. This method works fine for the operator since a human's deductive ability can discern, for instance, that the lighter colored linear feature up ahead is the road and not an obstacle.

In this example the sensor used no information (besides color) from the environment. The image generator simply displayed the pre-determined colors based on the stored values. Instead of the sensor detecting or measuring something as an input to use to compute the output, the image generator "sensed" the use of the NVG sensor and displayed a stored output. The Data Model did not allow the NVGs (or any sensor) to use the physical properties that can exist in the synthetic environment to determine how to provide an output.

2.5 The Need for Sensor Representation in SEDRIS

The need is fairly straightforward. Sensor integration is key to achieving realism in simulations. Simulations and simulators need to interoperate for training effectiveness and economical reasons. To achieve interoperability, individual simulation systems must interchange synthetic environment data as a pre-condition to operating in the "same" environment. Of the methods available, the most accurate, complete, and unambiguous manner for achieving successful data interchange rests with the SEDRIS project. Since the use of SEDRIS may become the preferred data interchange method in the M&S industry, it is imperative for SEDRIS to thoroughly represent sensors. The goal is for SEDRIS to support the *full* range of M&S applications, but the current Data Model (1.04d version) does not sufficiently include sensor representation.

Currently, there are many sensor simulations and models that use environmental factors to produce outputs and make decisions, and it is likely that there will be even more of them in the future. But they will be effectively useless if they cannot be shared

in a manner that is both affordable and reliable. Most SE databases being used today for SEDRIS interchange experiments contain sensor-related data, but that data is unable to be included in a transmittal since the Data Model is not yet complete regarding sensors. This research centers around the question: How can the SEDRIS Data Model be extended to include sensor representation?

This development effort focuses on both illuminating issues surrounding this question and attempting to answer the question. To accomplish this endeavor, the research examines the following research questions:

1. Why did the SEDRIS Team decide that using the Data Table class was the most efficient and unambiguous method for SEDRIS to support sensor simulation data using the current (1.04d) SEDRIS Data Model structure?
2. What sensor input requirements need to be represented in SEDRIS?
3. Can these common sensor simulation input requirements be mapped into SEDRIS?
4. Can a list of "actions required" be identified that will be useable by the SEDRIS Team to modify the Data Model to include the desired sensor input requirements?
5. Can a minimal sensor database interchange experiment using a modified Data Model be conducted to demonstrate if the data interchange was both loss-less and accurate?

For each of these research questions, several sub-questions were developed. By answering the sub-questions, the research questions are able to be answered. Each research question directly links to a step in the methodology. Chapter 3 details the step-by-step process used in this research effort to extend the SEDRIS Data Model to include sensor representation.

CHAPTER 3

SENSOR DATA MODEL DEVELOPMENT FOR SEDRIS

3.1 Introduction

The primary goal of this research effort is to examine and extend the *capability* of the current SEDRIS Data Model to fully include sensor representation. Based on the qualities of specific sensor simulations being used today (bounded by two dimensions; five M&S functional areas and four sensor categories), a generic list of sensor-related environmental attributes was created. Using these attributes, that characterize the population of sensor simulations and tools within the problem space, modifications to the SEDRIS Data Model are recommended. This chapter discusses the general method used to complete this research. It includes how the study of the current SEDRIS Data Model structure was approached, the study of sensor simulations being used today, and the reasons for using the set of sensor simulations/tools that were selected. This chapter will describe the methods and/or procedures that were followed: to map the sensor capabilities to the SEDRIS Data Model; to analyze the resulting mapping document; to verify and implement changes to the Data Model; and to demonstrate the modified SEDRIS Data Model.

There are a few conventions in use throughout this paper. References to the *current* Data Model are citing the 1.04d version released on November 25, 1997. The

modified Data Model, however, refers to the 2.0 version released on January 7, 1999. When referring to the Data Model and Data Dictionary, the first letter is capitalized to avoid using the acronym SEDRIS each time. Likewise, SEDRIS classes are discussed, the first letter of the class name will be capitalized, and in some cases, the class name will be enclosed in pointed brackets (e.g., <Example Class Name>) when the sentence context is difficult to understand without the brackets.

The SEDRIS project evolved rapidly throughout this research's timeline. To establish a baseline for standardization throughout this paper, the 1.04d Data Model release is used as the version from which initial research findings are cited. As the research progressed through the methodology to the point where recommendations were submitted for Data Model modifications, the Data Model had matured greatly in terms of functionality and presentation format. The majority of the changes were eventually incorporated into the first public release of the Data Model, called version 2.0 and formally re-titled as the SEDRIS Data Representation Model. Likewise, the minimal sensor database interchange experiment that was conducted to evaluate the modified Data Model used the version 2.0 Data Model release. For clarity, the term *Data Model* will be used throughout this paper to identify what is now called the SEDRIS Data Representation Model.

3.2 Current SEDRIS Data Model Structure for Sensors

As previously stated in chapter 2, the Data Model was immature with respect to sensors and did not fully include sensor representation. Additionally, some examples

why this was true were provided. In this section, a description of exactly how the current Data Model handles sensors and where they currently reside in the Data Model is presented. The intent is to relate the pertinent information about the Data Model without the discussion focusing on the computer programming aspects and complicated coding processes involved in SEDRIS.

When this research began with the SEDRIS development team in January 1998, the current Data Model version in use was 1.04d (shown in Appendix B). In the current Data Model, sensors were represented by the presentation domain attribute, called SE_PRESENTATION_DOMAIN_ENUM. According to the SEDRIS Data Dictionary (1997), this enumeration (enum) indicates the intended use of a Color, a Color Table, or a Geometry object with Rendering Properties. Figure 3.2.1 shows a small portion of the Data Model that illustrates the relationship between these classes.

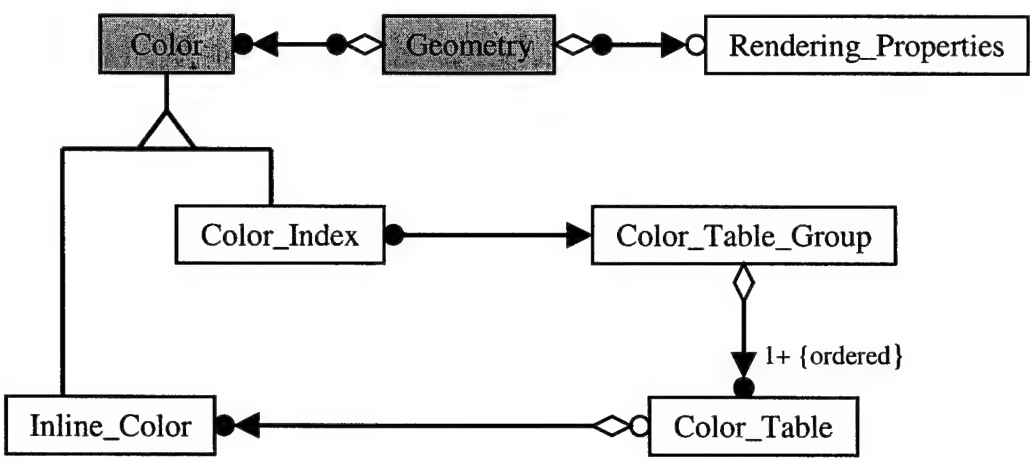


Figure 3.2.1. Relationship of Classes with SE_PRESENTATION_DOMAIN_ENUM (SEDRIS Data Model, 1997)

The Color class is used as an example to explain the relationships. The Data Dictionary (1997) defines the abstract class Color as the base class for colors in SEDRIS. As Figure 3.2.1 shows, Colors can either be <Inline Color> or <Color Index>. An <Inline Color> can have up to four <Color Data> (not shown in Figure 3.2.1). A <Color Index> references a <Color Table Group> from the <Color Table Library> (also not shown in Figure 3.2.1). Each <Color Table Group> has one or more ordered <Color Tables> which hold a list of <Inline Colors>.

Color has two attributes. Colors have a color_mapping attribute that defines how they are applied to the objects that use them. Colors also have a presentation_domain attribute in order to identify the type of sensor for which the color is appropriate. This presentation_domain can be overridden by using <Color Index> to refer to a <Color Table Group> which can have <Color Tables> with many different presentation_domains. The Data Dictionary (1997) further explains these relationships. A <Color Table Group> is an interchangeable group of one or more <Color Tables>. The first <Color Table> in the group is the primary <Color Table>. When a reference is made to a <Color Table> from somewhere in the transmittal (e.g., from a <Color Index> component of a <Polygon>), the reference identifies the <Color Table Group> and the index_number (the row number within the <Color Table>). An example is a case where there is a <Color Table Group> with two <Color Tables>. One <Color Table> with a presentation_domain attribute for normal, Out the Window (OTW) viewing, and another <Color Table> with a presentation_domain attribute to change the appearance of the normal view to be a view as seen through NVGs (p. 82).

The Data Dictionary (1997) lists the type definitions for SE_PRESENTATION_ DOMAIN_ENUM attribute as follows:

```
typedef enum
{
    SE_OTW,              /* out the window – human visual sensor */
    SE_IR_HI_BAND,       /* 8-12 microns */
    SE_IR_LOW_BAND,      /* 3-5 microns */
    SE_NVG,              /* Night Vision Goggles */
    SE_DAY_TV_COLOR,     /* Color TV */
    SE_DAY_TV_BW,        /* Black and White TV */
    SE_RADAR,            /* General Radar display – not concerned
                         with scan format */
    SE_SAR,              /* Synthetic Aperture Radar */
    SE_THERMAL,          /* thermal */
} SE_PRESENTATION_DOMAIN_ENUM;
```

This means that a Color class, a Color Table class, or a Geometry object through the Rendering Properties class can have a presentation_domain attribute that changes the color to represent any of the sensor types listed above. This study revealed that sensors *were* being represented in SEDRIS. But, as previously mentioned, two things could be concluded:

- The visual representation method supported in the current Data Model is considered non model-related since only the visual "mode" is changed based on the sensor type being used. This method does not support a SE that captures the measurements of physical properties in the environment that sensor models use as input data.
- The sensor types captured by the presentation_domain attribute do not *fully* represent the diversity in the sensor simulation population.

3.3 Using the Data Table Class to Support Sensor Data

The first research question to answer was why the SEDRIS Team decided that using the Data Table class was the most efficient and unambiguous method for SEDRIS to *fully* represent sensors of both model-related and non model-related simulations using the current Data Model structure. The SEDRIS Team members made this decision during January 1998, the same month this research started. Although this research effort had made no direct impact at this stage, it was clear that this decision was the first step towards fully representing sensors in SEDRIS, and therefore warranted research. The results of the research explaining the decision by the SEDRIS Team to use the Data Table class, an existing class in the Data Model structure, are presented in chapter 4. In section 4.2 of chapter 4, the findings are discussed for both the multiple organizational methods for sensor data in general and the organizational method selected for use in SEDRIS. Additionally, there is a discussion on two alternative methods to capture sensor data in the Data Model, an investigation of how and why the SEDRIS team selected the Data Table class as the preferred method, and a description of the Data Table class in detail.

Once it was understood why the decision was made to use the Data Table class for capturing sensor data in SEDRIS, the next step was to determine what sensor input requirements needed to be represented in SEDRIS.

3.4 Study of Sensor Simulations

To answer research question 2, what sensor input requirements needed to be represented in SEDRIS, a study of existing sensor simulations was started. The intent

was to get an idea of the types of sensor simulations currently being used, their characteristics and capabilities, and whether they are considered model-related or non model-related. For reasons discussed in chapter 1, the sensor simulations that make up the population of concern are all military-oriented sensor simulations and tools. Since it is not possible to study every simulation in the population, a representative sample needed to be selected.

To accurately represent the population, the sample has to cover the diversity that exists in the population. That means the sample has to adequately represent sensor simulations/tools from both the broad M&S user community and those that operate throughout the EM spectrum. According to Piplani, Mercer, and Roop (1994), the M&S user community is broken down into five functional areas: (1) education, training and operations; (2) research and development; (3) test and evaluation; (4) analysis; and (5) production and logistics (p. 2-2). Each of these functional areas encompasses many M&S applications. Table 3.4.1 lists specific applications for each of the functional areas. An assumption was made that by selecting sensor simulations/tools that represent the five functional areas, an adequately diverse sample could be obtained from the broad M&S user community.

Recalling from chapter 1, sensors that operate throughout the EM spectrum are classified in four categories: optical systems, lasers, thermals, and radars. Selecting sensor simulations/tools that both model sensors from these four sensor categories and pass the "interoperable test" ensures that a diverse sample along the EM spectrum will be obtained. To find candidate simulations that met the requirements, recommendations of

subject matter experts as well as popular sensor simulations/tools were considered. Of equal importance was the simulation's availability. To answer the research questions, the information about the candidate sensor simulations/tools had to be both obtainable within the research horizon and meet budgetary constraints.

Table 3.4.1

Specific Applications for M&S Functional Areas

M&S Functional Area	Applications
Education, Training, and Operations	Re-creation of historical battles Doctrine and tactics development Command and unit training Operational planning and unit rehearsal Wartime situation assessment
Research and Development	Requirements definition Engineering design support Systems performance assessment
Test and Evaluation	Early operational assessment Development and operational test design Operational excursions and post-test analysis
Analysis	Campaign analysis Force structure assessment System configuration determination Sensitivity analysis Cost analysis
Production and Logistics	System producibility assessment, Industrial base appraisal, and Logistics requirements determination

(Piplani et al., 1994)

A sample of 11 sensor simulations/tools was identified for detailed study. The names and a brief description of each of the 11 simulations/tools are listed below:

- Close Combat Tactical Trainer (CCTT) is the U.S. Army's system of manned armored vehicle simulator modules and workstation that train collective armor and infantry tasks. The CCTT includes sensors that are critical for successful training experience. Project managed by STRICOM. Analysis of the CCTT database provided by Evans and Sutherland, Inc.
- IRGen® Infrared Data Base Modeler from Technology Service Corporation is an IR signature modeling tool that creates an IR file that is compatible with many real-time graphics generators.
- Camber Radar Toolkit™ is the Camber Corporation's comprehensive real-time radar simulation that provides the software required to build a Ground Mapping Digital Radar Landmass Simulation. This allows the modeling of realistic radar landmass returns and shadows for any geometry of the simulated radar and the environment.
- RadarWorks™ from Photon Research Associates, Inc., provides quantitative, deterministic radar scene simulation. It is a commercial off-the-shelf radar scene simulation package providing pixelized Radar Cross Section maps and realistic displays of various radar imaging modes in real-time.
- Sensor Texture Maps (STM) from Photon Research Associates, Inc., provides texture map products for use in high-fidelity sensor simulation. Sensor Texture Maps enable existing image generation systems to simulate correlated out-the-window and sensor views such as electro-optical, infrared, radiance or radar backscatter views.

- Thompson Training & Simulation (TTS) uses an internal format called BDD3. TTS has many training simulations that use radar and infrared sensors for military and civilian applications.
- RADSIM™ from Science Application International Corporation (SAIC) is a multi-mode radar simulation capable of supporting a variety of airborne radar systems. Its primary focus is in air-to-ground and air-to-air operating modes.
- Irma is the standard Air Force tactical weapons multi-sensor target and background signature prediction model used primarily for research and development. It is being maintained and enhanced by Nichols Research Corporation under the Multi-Sensor Modeling and Analysis program to the Munitions Directorate of the U.S. Air Force Research Laboratory.
- SensorVision™ from Paradigm Simulation is a software toolkit that computes and displays quantitative sensor-view images of any environment containing natural backgrounds, cultural features, and entities. It supports the simulation of sensor systems operating at any wavelength, from visible through infrared.
- Special Operations Forces Aircrew Training System (SOF ATS) Data Base Generation System (DBGS) from Lockheed Martin Tactical Defense Systems (LMTDS) is fielded at the SOF ATS facility at Hurlbert Field in Florida. The DBGS has the capability of assembling a simulation database for any area of the world from whatever data may be available for that area. The resulting simulation database is used during mission rehearsals and training scenarios.

- "Paint-the-Night" is a thermal scene generation simulation that provides realistic electro-optic simulations for DIS exercises and is used extensively for research and development. It was developed by the U.S. Army Communications & Electronics Command's Night Vision and Electronic Sensors Directorate (NVESD) at Fort Belvoir, Virginia, and the Army Research Laboratory at Aberdeen, Maryland, working cooperatively with private industry.

Table 3.4.2 shows how the sample that was selected covers both the sensor aspects of the M&S community's five functional areas and the four sensor categories that represent the EM spectrum. The five functional areas are listed down the left and the four sensor categories are along the top of the table. The cells within the body of the table list the names of the corresponding simulations/tools that were just described above. As Table 3.4.2 indicates, the sample that was selected contains at least one and at most seven sensor simulations/tools that cover both the five M&S functional areas and the four sensor categories. Based on this coverage and the research constraints, it was concluded that these 11 simulations/tools represented an adequately diverse sample for this study. Once satisfied with the sample, a detailed study of each sensor simulation/tool was conducted to determine the sensor input requirements needed to be represented in SEDRIS. The results of the detailed sensor simulation/tool study and the findings of whether the sampled simulations/tools were model-related or non model-related are summarized in chapter 4.

Table 3.4.2

Sensor Simulation/Tool Coverage of M&S Functional Areas and Sensor Categories

M&S Functional Areas	Optical Systems	Lasers	Thermals	Radars
Education, Training, & Operations	CCTT IRGen® STM TTS BDD3 SensorVision™ SOF ATS Paint-the-Night	CCTT SensorVision™	CCTT IRGen® STM TTS BDD3 SensorVision™ SOF ATS Paint-the-Night	Radar Toolkit™ IRGen® RadarWorks™ STM TTS BDD3 RADSIM™ SOF ATS
Research & Development	IRGen® STM Irma SensorVision™ Paint-the-Night	Irma SensorVision™	IRGen® STM Irma SensorVision™ Paint-the-Night	Radar Toolkit™ IRGen® STM RADSIM™ Irma
Test & Evaluation	IRGen® STM Irma SensorVision™ Paint-the-Night	Irma SensorVision™	IRGen® STM Irma SensorVision™ Paint-the-Night	IRGen® STM Irma
Analysis	STM Irma Paint-the-Night	Irma	STM Irma Paint-the-Night	Radar Toolkit™ STM Irma
Production & Logistics	IRGen® Irma	Irma	IRGen® Irma	IRGen® Irma

3.5 Mapping Sensors Simulations to SEDRIS

The results of the detailed study of the 11 sensor simulations/tools, which is presented in chapter 4, yielded the sensor-related environmental properties (input

requirements) that SEDRIS needed to capture in the Data Model. After recording the input requirement data, work was started on research question 3; can the common sensor input requirements (the sensor-related environmental properties) be mapped into the SEDRIS Data Model. To answer this question, a mapping document was written that attempts to match the properties from the sampled sensor simulations/tools to SEDRIS attributes. First all the sensor simulations' capabilities and characteristics were listed and the sensor-related environmental properties were grouped where a physical commonality exists. Next, for each sensor simulation characteristic and capability, a determination was made whether or not there was an existing counterpart in the current Data Model. When there was no counterpart in the Data Model, a corresponding SEDRIS attribute was created. Lastly, the mapping document was written. Details concerning the development process of the mapping document and a sample from the completed mapping document are covered in chapter 4.

3.6 Developing the List of "Actions Required" to Modify the Data Model

An analysis of the completed mapping document was completed to answer research question 4; can a list of "actions required" be identified that will be useable by the SEDRIS Team to modify the Data Model to include the desired sensor input requirements. The mapping document development process should result in a list of *new* sensor-related SEDRIS attributes. Additionally, through the mapping document analysis and discussion with sensor and SEDRIS subject matter experts, a determination is made whether other sensor-related aspects are needed in the Data Model for proper functioning.

The combination of these findings yields what actions are necessary to modify the Data Model for full sensor representation. The "actions required" are expected to fall into several common data model categories (e.g., create a new data table type, create a new data table label, etc.). The information is compiled into a list of "actions required" that the associates on the SEDRIS Team can use to modify the Data Model. This step in the methodology addressed research question 4, and ends the analysis phase of the research. The development of the "actions required" list is discussed in detail in chapter 4.

3.7 Modification of the SEDRIS Data Model

This section outlines the process by which all modifications are made to the Data Model to include the desired sensor-related environmental attributes. Once the list of "actions required" is finalized from the analysis phase, the research enters the implementation phase of the study. This phase starts with a peer-review process and culminates with the changes being programmed into the Data Model.

When the SEDRIS project was started, a core team of individuals with extensive experience in developing simulation databases was assembled to draft the first Data Model. The core team solicited industry review to gather insight about the Data Model's use. Industry participants were selected from responses to a Broad Agency Announcement process. The industry participants were directed to use and evaluate the Data Model. A broad spectrum of data providers, database developers, and applications developers was selected to ensure that the Data Model was complete and capable of supporting a wide variety of environmental representations and vendor unique database

designs (Foley et al., 1998). An associate developer relationship was implemented to achieve maximum leverage of contractor experience in recommending changes to the Data Model. When this research began, there were three DoD sponsoring agencies and 13 SEDRIS associate contractors. At publication time, shortly after the 2.0 version of the Data Model was released, there were 24 SEDRIS partners. Appendix C provides more information on the SEDRIS Team.

Prior to submitting any changes to the Data Model, the issue in question is discussed by those associates who will be affected by the changes. This peer-review process is accomplished either as an agenda item during one of the periodic SEDRIS Associate Meetings or discussed using one of the focused electronic mail reflectors. The specific issue is prepared in a standardized SEDRIS Change Request (SCR) format. The author of the SCR is usually the subject matter expert in the area that needs attention. Regardless of the discussion venue, when a consensus is reached between all affected associate contractors, the final SCR is generated and submitted to Science Applications International Corporation (SAIC) for inclusion in the next release of the SEDRIS Data Model and Data Dictionary. This same procedure continued until SEDRIS released the public 2.0 version of the Data Model and Data Dictionary in January 1999. During its development, the Data Model was updated as often as needed, which at some points was as frequent as weekly. Following the public release, Data Model updates are scheduled to occur about every six months.

The Data Model changes recommended to the SEDRIS Team as a result of this study will follow the same implementation procedure outlined above. In chapter 4, the

changes made to the Data Model based on the mapping document analysis are discussed. As part of the explanation in chapter 4, examples are provided of the SCRs that were submitted during this implementation phase of the study.

3.8 Demonstration of the Modified Data Model

The last step in the methodology was the test phase of the study. It attempted to answer research question 5; can a minimal sensor database interchange experiment using a modified Data Model be conducted to demonstrate if the data interchange was both loss-less and accurate. A demonstration using a minimal sensor database interchange experiment provided insight into the analysis and implementation phases of this research. Using the experiment's results, conclusions were drawn about the validity of the sensor portion of the modified Data Model and recommendations for further changes to the Data Model were made. The demonstration was conducted using the 2.0 version of the SEDRIS Data Model, which is referred to as the *modified* Data Model.

3.8.1 Full Database Interchange Experiment Description

The ideal evaluation would be a full database interchange experiment using either a simulation database containing its imbedded sensor information (e.g., CCTT) or a stand-alone sensor simulation database (e.g., Irma) selected from the original sample. The basic task would attempt to interchange an entire database from a producer's native format into a consumer's differing runtime application format using SEDRIS as the

interchange specification. The test would compare the database content to determine if the data interchange was loss-less and accurate. This process is graphically depicted in Figure 3.8.1.

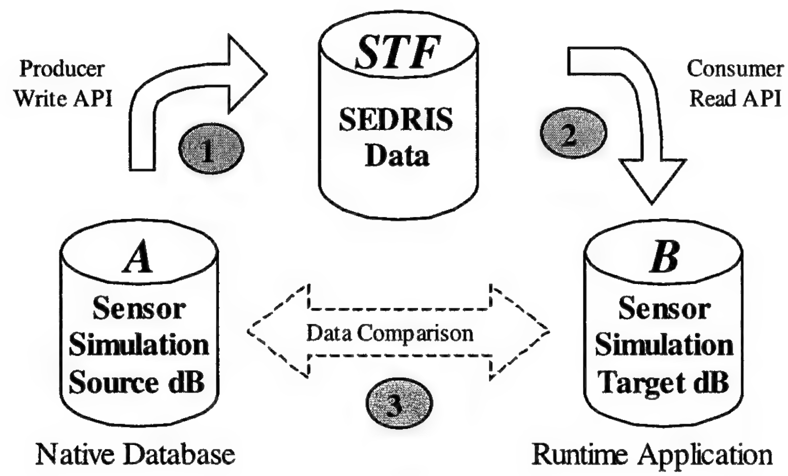


Figure 3.8.1. Ideal Database Interchange Experiment

Although the final comparison between databases *A* and *B* is of primary concern (step 3 from Figure 3.8.1), a data comparison would be conducted in two stages. The first stage data comparison would take place after the native database is converted into the SEDRIS data format (formally called SEDRIS Transmittal Format, or STF) using the Producer's Write API (step 1 from Figure 3.8.1). The data comparison would check if the conversion process is loss-less (e.g., no missing data or lost data) as well as if the resulting data captured in SEDRIS is accurate (e.g., no change in the data values or no significant rounding errors). For acceptability purposes, each producer and consumer would have to set their own measurement tolerances. The second stage data comparison would take place after the SEDRIS data in STF is converted into the consumer's runtime

data comparison would check to determine if the conversion process is loss-less and accurate.

One of the SEDRIS tools that could be used to compare databases *A* and *B* above during both stages is the Side-by-Side Terrain Viewer developed by AcuSoft, Inc. According to STRICOM (1998b), this tool "supports the side-by-side viewing of two SEDRIS transmittals to visually compare their terrain elements. Identified differences can be made to conform by replacing the data in one transmittal with data from the other transmittal" (p.23). The Side-by-Side Terrain Viewer allows a "fly through" visual comparison of up to 16 separate simulation databases simultaneously on one system/monitor. According to Jesse Liu, President of AcuSoft, Inc., this viewer tool was originally intended to visually compare traditional terrain databases, but it can be easily adapted to compare sensor-related environmental properties in environmental databases as well (J. Liu, personal communication, February 11, 1998). Additionally, the suite of available SEDRIS software tools and utilities complements the Side-by-Side Terrain Viewer for database comparison purposes.

3.8.2 Minimal Database Interchange Experiment Description

Based on resource constraints in terms of time available and funding, a full interchange experiment was not within the scope of this study. Instead a minimal interchange experiment was conducted using a test database. The minimal database, constructed by Russ Moulton, Jr., of JRM Enterprises, Inc., used sample data from the "Paint-the-Night" (PTN) IR sensor simulation. In order to keep the test and resulting

"Paint-the-Night" (PTN) IR sensor simulation. In order to keep the test and resulting analysis under control, the test database consisted of a PTN sample polygon data set and its associated thermal system. The producer built the sample PTN data set so it contained multiple sensor-related environmental attributes. Although it is small, this minimal database represented the producer's full database in the native format. In order to eliminate the cost that JRM Enterprises, Inc., would incur to build a Producer Write API (step 1 of Figure 3.8.1), an alternate approach was needed.

The Producer Write API step of an ideal interchange experiment was replicated by a two-step procedure. In the first step, the minimal PTN database was mapped to the modified Data Model from a *description of the data* to create an object diagram. There is a subtle, yet important distinction to note here. In the case of the ideal database interchange experiment, the Producer Write API works directly on the native database files and the data values (instead of a textual description of those values). The minimal PTN database was a one-polygon database with multiple sensor-related environmental attributes described by exact data values. Using the textual description, an object diagram was built using the Data Model to capture a one-polygon database's structure, sensor-related environmental attributes, and data values. The resulting object diagram was a representation of the PTN data in terms of instances of corresponding SEDRIS Data Model classes complete with the legal entries in the required fields.

In the second step, Bill Horan, a Simulation Engineer at SAIC, used the object diagram as the source document to create a SEDRIS transmittal. He constructed the SEDRIS database transmittal using a special software utility he designed that performs a

producer's format, called *A*, and a STF database file that is based on a description of the producer's native database. By constructing the SEDRIS database transmittal from the object diagram (based on the PTN minimal database), Bill Horan removed the funding obstacle concerning the development of the Producer Write API to convert the PTN data into SEDRIS data.

The SEDRIS data in STF was then converted into the target database format, which happened to be the same as the PTN native database format, using a Consumer Read API. Due to time and funding constraints, this API was written by Bill Horan of SAIC. The resulting database, called A' (*A-prime*), represented a consumer's differing runtime format. When that step was complete, the two databases (A and A'), were ready for the data comparison step. Figure 3.8.2 depicts the minimal database interchange experiment process.

Although the producer's native PTN database format and the consumer's runtime PTN format were the same in this case, the two data files (A and A') had different origins. The difference rests with the fact that the A data originated from the actual PTN database, while the STF data was created by a database generation program based only on a description of the native PTN data. Therefore, when the STF data was converted into the A' consumer PTN format, the data originated from a different source. This slight distinction between A and A' represents the need for the data comparison step in Figure 3.8.1 of databases A and B.

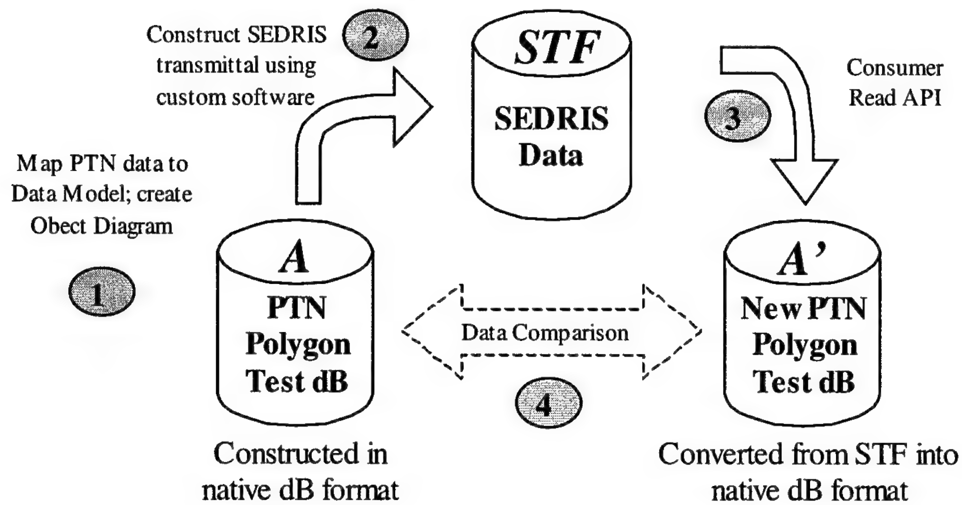


Figure 3.8.2. Minimal Database Interchange Experiment

3.8.3 Data Comparison Procedure

Similar to the case of the ideal interchange experiment, the data comparison step of the minimal database interchange experiment (step 3 of Figure 3.8.2) was accomplished in two stages. The first stage data comparison took place after database *A* was converted into STF data (following the completion of steps 1 and 2 of Figure 3.8.2). The second stage data comparison took place after the STF data was converted into *A′* data (following the completion of step 3 of Figure 3.8.2). Both stages checked if the data interchange was loss-less and accurate.

A *loss-less data interchange* is considered a success if no data is lost or is determined to be missing upon data comparison. In the first stage data comparison, the loss-less goal is for every data value in *A* is to have a unique storage location in STF. In the second stage data comparison, the loss-less goal is to have the same number and type

of data values in both A and A'. For example, if A has 50 data values that are numerical, then the goal is for A' to have 50 data values that are numerical. In the case of the minimal interchange experiment, it was expected that the data comparison of A and A' would be a success in terms of the loss-less goal.

An *accurate data interchange* is considered a success if no data is changed in any way during the data interchange process, including rounding errors. In both data comparison stages, the accuracy goal is for every data value to have the exact same value before and after the conversion. For example, 0.5 and 0.5000 are the exact same value whereas 0.5 and 0.4999 are not the exact same value. As mentioned earlier, each producer and consumer would have to set their own tolerances for data comparison acceptability purposes. For instance, some database comparisons may have their metric tolerances set to accept a difference between 0.5 and 0.4999 as equivalent values if that level of rounding error would not affect the simulation outcome. In the case of the minimal interchange experiment, however, the data comparison tolerance was set to reject *any* difference in data value. It was expected that the data comparison of A and A' would be a success in terms of the accuracy metric.

Since the minimal database interchange experiment was small in terms of the number of data values for data comparison, a SEDRIS tool such as the Side-by-Side Terrain Viewer was not used for data comparison. Instead, a data comparison between A and A' was conducted by manually comparing each data value in both stages. The details of the conduct of the experiment and the results of the two-stage data comparison are discussed in chapter 4.

CHAPTER 4

RESULTS OF THE SENSOR DATA MODEL DEVELOPMENT

4.1 Introduction

This chapter presents the results found during the implementation of the methodology in chapter 3. Each section in chapter 4 basically mirrors a section in chapter 3. First it provides the results of the study of the SEDRIS Team's decision to use the Data Table class for representing sensors, the results of the detailed study of the sample of 11 sensor simulations/tools, and the results of the mapping document development process. Then, the focus shifts to the analysis of the mapping document that resulted in the list of "actions required" to modify the Data Model, the presentation of the changes that were recommended to the SEDRIS Team, and the discussion of the results of the minimal database interchange experiment.

4.2 Research on the Decision to Use the Data Table Class

As mentioned in chapter 3, the SEDRIS Team's decision to use the Data Table class concerning sensor representation warranted research because it was the first step towards fully representing sensors in SEDRIS. The first step in this research focused on determining how sensors could be categorized. One of the SEDRIS project's advantageous characteristics is the ability to organize data in multiple ways. According

to Long Nguyen, a sensor expert working at the Naval Air Warfare Center – Training Systems Division (NAWC-TSD) in Orlando, FL, sensors can be organized in the following manners:

- By type (e.g., radar, laser, etc.),
- By band (e.g., UHF band for radio, S band for radar, IR band for infrared, etc.),
- By platform (e.g., M1A1 tank, F16 fighter jet, AN/PVS-7A night vision goggles, etc.), or
- By the physical properties they measure in the environment (e.g., reflectivity, transmission loss, solar irradiance, etc.).

(L. Nguyen, personal communication, July 28, 1998).

When the study of the existing Data Model structure began, Long Nguyen discussed these different sensor organizational methods and the impacts of using them for representing sensors in SEDRIS. Table 4.2.1 highlights the results of the research regarding the advantages and disadvantages of the different sensor organizational methods.

Table 4.2.1

Methods For Organizing Sensor Data and Their Impacts

Sensor Organizational Method	Impacts: Pros	Impacts: Cons
1. By Type (e.g., thermal imager, laser range finder, laser target designator, synthetic aperture radar, target tracking radar, FLIR, NVG, direct view optics, etc.)	- Easy to use by consumers, if they know the type of sensor used in the interchanged database and it matches the requirements they need.	- Requires a class for every type of sensor (see Figure 1.4.2 for common types). More than 50 new classes. - Classes would change as the types of sensor changes. - No standard naming convention for sensor types in the sensor community
2. By Band (e.g., LF, HF, VHF, UHF, Microwave, P-band, S-band, X-band, K-band, IR, Far IR, Near IR, SWIR, LWIR, UV, X-ray, etc.)	- More specific than 'by type' because it narrows the type of sensor to where it operates along the EM spectrum.	- Requires less classes than 'by type' because there are fewer bands than types. More than 25 new classes. - Bands with different names overlap on EM spectrum. No standard naming convention for all bands, therefore ambiguity. - Classes would change as sensors are added.
3. By Platform (e.g., AN/PVS-4 Night Sight for the M16 series rifle, AN/APG-169 Attack Radar System in the F-111C aircraft, etc.)	- Very specific since each sensor has a different nomenclature based on the platform on which it resides. - Easy to use by consumer.	- Requires a huge number of classes, one for every single sensor nomenclature. More than 300. - Classes would constantly change as new vehicles and weapon systems are fielded
4. By Physical Properties Measured (e.g., reflectivity, emissivity, thermal conductivity, solar absorptivity, etc.)	- Most specific method since every sensor measures physical properties in the environment. - Requires no new classes. Can use the existing Data Table class.	- Not as easy to use by consumers since they have to know the properties that are being measured.

4.2.1 Defining Efficient and Unambiguous

In order to understand the SEDRIS Team's decision, an understanding of the definition of efficient and unambiguous was needed. The *most efficient method* is defined as the single method or the combination of methods in which the fewest new classes had to be added to the current Data Model. Having the fewest number of classes in the final Data Model means the SEDRIS Team needs to expend less time, money and frustration during the Data Model and Data Dictionary creation, verification, and maintenance (W. H. Horan, personal communication, July 30, 1998).

The *most unambiguous method* definition is a more subjective measure, but easily recognizable using common sense. It is defined as the method that allows a database producer and/or consumer to describe the SE using terms with the least number of possible meanings. For example, a thermal imaging sensor can be described with varying levels of ambiguity. From least to most unambiguous, the thermal imaging sensor could be described as (1) an infrared sensor (by type), (2) a sensor that is in the SWIR band (by band), (3) a Tank Thermal Sight (TTS) on the M1A2 Tank (by platform), or (4) a sensor measuring physical properties such as reflectivity, emissivity, and absorptivity (by physical properties measured).

Using these definitions and the information in Table 4.2.1, the two methods that are easily ruled out are the "by type" and "by platform" methods. Trying to include the model in SEDRIS for every sensor type or sensor platform is not practical. Just compiling the list of every sensor type and/or sensor platform and nomenclature would be a monumental undertaking. However, if such a list were compiled, it would be a constant

battle to keep the list up to date in today's growing technology market. Using the efficient and unambiguous metrics, these two methods ("by type" and "by platform") would require the creation of the most new data classes (therefore, not efficient) and they are not the most unambiguous options available.

Since SEDRIS aims for an unambiguous description of the complete environment, and all sensors operate in a band and measure physical properties in the environment, the organizational method selected was a skewed combination of the "by band" and "by physical property measured" methods. By recalling the definition of a sensor from chapter 1, it is indisputable that all sensors measure physical properties. Using the method of "by physical property measured" as the primary organizational method and the "by band" information as a secondary method, the SEDRIS Team thought that a generic representation of all sensors could be achieved without regard to the particular nomenclature or type of the sensor system. In effect, this method supports the creation of a place in the Data Model for every possible physical property measured by a sensor and every type of standard frequency band used in the sensor community. And, when using SEDRIS as an interchange specification, a SE database producer or consumer's goal is to find a place in the Data Model for every piece of their data.

4.2.2 Alternate Methods to Capture Sensor Data

Having decided which method to use to organize sensor data in SEDRIS, Long Nguyen determined that there were two ways to create the "spots" (i.e., capture the sensor data) in the Data Model. Either (1) create a new class for each sensor-related

environmental property and aspect or (2) use the existing Data Table class (L. Nguyen, personal communication, January 15, 1998). These two methods were one of the topics discussed during the SEDRIS Associates Meeting (SAM) #6 on January 13, 1998.

Although every sensor-related environmental property and aspect had not yet been enumerated, the number was large. At that time, the 1.04d Data Model already contained 290 classes (W. H. Horan, personal communication, July 30, 1998). The sensor discussion at SAM #6 concluded that option (1), adding a new class for every sensor-related environmental property and aspect, would make the Data Model overbearing and complicated.

For example, if every polygon or vertex had several sensor-related properties, that polygon or vertex would have to be associated with every one of the respective sensor-related property classes. This conclusion ran contrary to the definition of the *most efficient method* discussed earlier. Additionally, this option does not agree with SEDRIS' Guiding Axiom #9, which states that "the implementation of the SEDRIS transmittal medium must not impose unreasonable resource demands and must be efficient in handling data" (STRICOM, 1998b, p. 4).

The other alternative, option (2), was to use the existing Data Table class in the Data Model. In general, data tables handle the sensor-related properties much more efficiently because of their multi-dimensional nature. Figure 4.2.1 illustrates where the Data Table class fits into the Data Model and how it relates to Geometry and Feature classes.

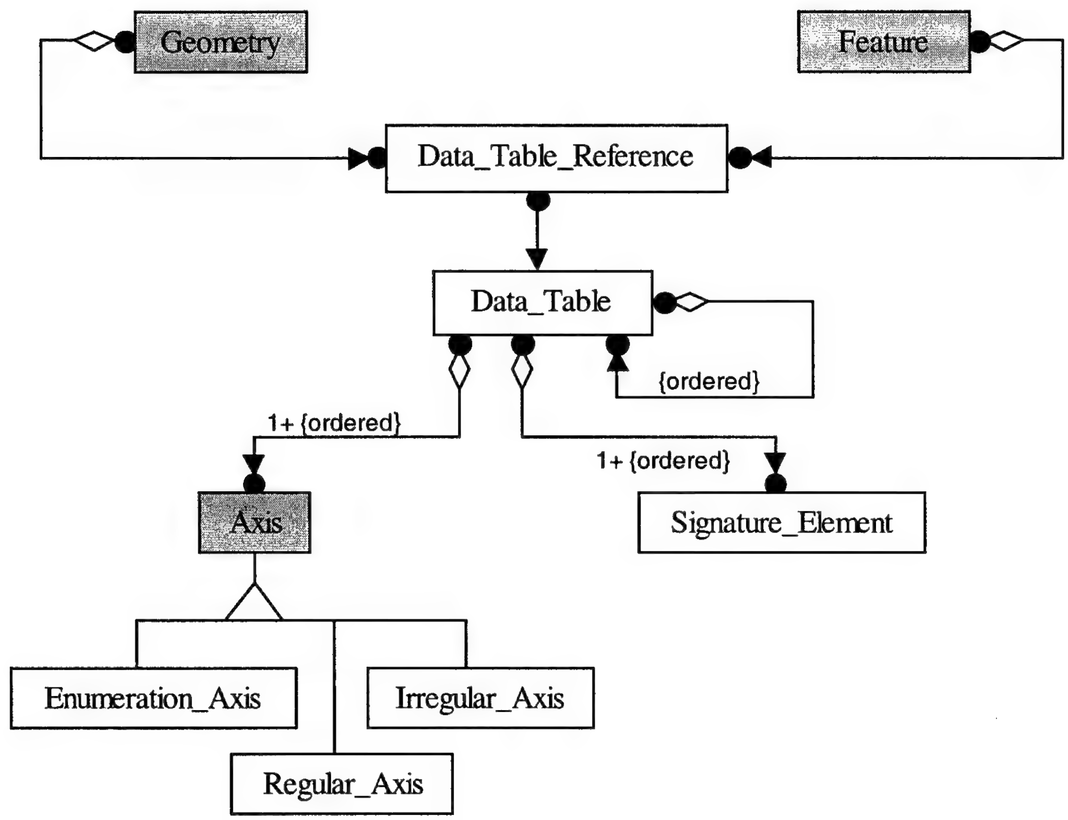


Figure 4.2.1. The Data Table Class Relationships

(SEDRIS Data Model, 1997)

As Figure 4.2.1 denotes, both a <Geometry> and <Feature> can have zero or more <Data Table References> that reference (is associated with) exactly one <Data Table>. A <Data Table> has one or more ordered <Axis> which are either an <Enumerated Axis>, a <Regular Axis>, or an <Irregular Axis>. The <Data Table> also has one of more ordered <Signature Elements>. According to the Data Dictionary (1997), a <Signature Element> has two attributes, value_type and value_label. The value_type (formally called SE_DATA_TABLE_VALUE_TYPE_ENUM) is a list of the

types of values that can be stored in a cell within a Data Table (e.g., a value of type Boolean, signed and unsigned integers, etc.). The value_label (formally called SE_DATA_TABLE_LABEL_ENUM) is the list of the types of values used to tell you *what* is being represented by a cell value in a Data Table (as used in the value_label field of a Signature Element object). This same list is also used to identify *what* an Axis is measuring (as used in the axis_type field of an Axis object). So the value_label enumeration tells you *what* a value means and the value_type enumeration tells you *how* a value is represented.

Most sensor simulations/tools fall along the fidelity continuum and use different methods to model the environment. Likewise, the properties (attributes) that sensors capture in the environment will be different. Further, each sensor-related property can be a function of many other properties. For example, a polygon may have the attribute of reflectivity, whose value is a function of its surface material type, polarization, wavelength, incident and reflected azimuth angle, and incident and reflected elevation angle. These attribute relationships are easily captured in *n*-dimensional data tables (i.e., data tables allow the use of as many axes as needed to define these relationships).

Since most database producer's formats are different, SEDRIS needs to provide an expandable, generic way to handle any possible combination of sensor-related properties in the environment. SEDRIS is not concerned with how the individual sensor simulation producers determine the attribute values for their SE or how the sensor computes its output during runtime. It is focused on the underlying static data that needs to be interchanged properly. As such, SEDRIS needs to provide a "spot" for all data

during the interchange process. The Data Table class gives the database producers and consumers the most flexibility for fully representing the SE. Therefore, the SEDRIS Team adopted the use of the Data Table class.

4.3 Results of the Study of Sensor Simulations

Once the reasons behind using the Data Table class for sensor representation were understood, the study of sensor simulations/tools was started to determine what sensor input requirements needed to be represented in SEDRIS. Recalling from chapter 3, a sample of 11 sensor simulations/tools was identified for detailed study that represented the diversity within the military-oriented sensor simulation community. Table 4.3.1 lists the characteristics of the sampled sensor simulations/tools. This table retains the same basic format as Table 3.4.2 from chapter 3. It keeps the same four sensor categories along the top, but lists the sensor simulations/tools down the left side. An additional column is included in this table to capture data on whether each of the simulations/tools was model-related or non model-related. The cells of the table describe the results of the detailed research of the sensor simulations/tools. Refer to the List of Acronyms at the beginning of this document to clarify any acronyms used in Table 4.3.1.

Table 4.3.1

Sampled Sensor Simulation/Tool Characteristics

Sensor Simulation/Tool	Model-related?	Optical Systems	Lasers	Thermals	Radars
1. CCTT	No	- Image intensifier - Direct view optics	- Laser range finder	- Tank Thermal Sight (TTS)	
2. IRGen®	Yes	- Low light level TV		- Creates IR texture maps	- MMW radiometer
3. Camber Radar Toolkit™	Yes				- RBGM - DBS - SAR - AGR - Aircraft detection & tracking
4. PRA RadarWorks™	No				- RCS maps - RBGM - DBS - SAR
5. PRA Sensor Texture Maps	Yes	- Out-the-window view - Night vision scenes		- Thermal imagers	- Any Radar
6. TTS BDD3 Radar Sim	No	- Optical sys - NVDs		- Thermal imagers	- Any Radar

Sensor Simulation/Tool	Model-related?	Optical Systems	Lasers	Thermals	Radars
7. RADSIM™	Yes				- DBS - SAR - TFR - Real beam imagery
8. Irma	Yes	- Visible scene generation	- Ladar	- Passive IR - Passive MMW	- MMW SAR - Real beam MMW
9. SensorVision™	Yes	- Any visible sensor system	- SWIR sensors - Laser rg finders	- Thermal imagers - FLIR sensors	
10. SOF ATS	No	- Out-the-window view - NVDs		- IR sensors	- Radar sensors
11. Paint-the-Night	Yes	- NVDs		- Thermal imagers	

The study of the characteristics of the sensor simulations/tools in Table 4.3.1 and the investigation of the underlying physical properties that these simulations/tools require during runtime yielded the sensor-related environmental properties that SEDRIS needed to capture in the Data Model. Early in this chapter, it was concluded that the best method to organize sensor data in SEDRIS was by using the "by physical property measured" as the primary organizational method. The underlying physical properties that were found during the research of the sensor characteristics in Table 4.3.1 formed the basis for the

changes to the Data Model. These details are part of the mapping document shown in the section 4.4.

4.3.1 Model-Related Versus Non Model-Related Sensor Simulations

Although finding the sensor-related environmental properties (input requirements) was the primary goal, another area of interest was in the number of sensor simulations/tools in the sample that were model-related versus not model-related. The results of the study revealed that the majority of the sampled sensor simulations/tools were model-related. Indeed, Table 4.3.1 indicates that 7 of 11, or 64%, of sensor simulations/tools from the sample were model-related. Recalling from chapter 1, sensor simulations that are model-related have the sensors interact with the environment, then use the information gained from the environment to calculate the sensor's output. These sensor simulations have a high measure of realism because they are modeled to replicate how the corresponding real-world sensor determines its output. Those sensor simulations that are not model-related, on the other hand, do not interact with the SE the same way. The geometries and features in the SE do not have sensor-related attributes. Instead, the most common technique is to alter the visual representation of the entire SE based on the viewing mode. These types of sensor simulations have a low measure of realism because they are not modeled upon the method used by their corresponding real-world sensors to determine their output.

It was anticipated that the findings would show that the majority of sensor simulations/tools were not model-related. This impression came from the initial audit of

the sensor simulations in the population when this research began in early 1998. The representation of synthetic environments for simulations seemed to revolve around the visual sector of the M&S community. In other words, most of the SEDRIS associates and SE database producers on the SEDRIS Team were focused on how to make the visual world more appealing and realistic to the human eye. A much smaller group of simulation scientists and engineers were concerned with realism that pertained to physics-based models of the SE, such as those working on CGF integration. These database developers focused on how to make the virtual world more like the real world using the physical properties that we know to exist, but that we cannot see visually. Therefore, using these experiences as the initial knowledge base, it was expected that the representative sample would show a majority of sensor simulations/tools to be non model-related.

4.3.2 Explanation of Model-Related Findings

How can the finding that the majority of the sample was model-related be explained? There are two possible explanations, and the explanations may not be mutually exclusive. The first explanation relates to how the sample was selected, which was primarily based on the recommendations of SMEs. First, the SMEs understood that the goal was to find a small number of sensor simulations that would adequately represent the (bounded) population of sensor simulations. Second, most sensor SMEs are from the group of simulation scientists and engineers that are concerned with physics-based models. Therefore they realized that in order to make a Data Model that would be

capable of interchanging complex physics-based databases, it should be based on sensor simulations that are model-related. So it is no surprise that they recommended mostly model-related sensor simulations/tools.

The second possible explanation is based on the time span of this research. Another factor on which the sample selection was based was popularity. As postulated in chapter 1, as the computational capacity of simulations and training systems continues to increase and the cost of purchasing more processing power decreases, a greater number of model-related sensor simulations are expected on the market. The research timeline spanned more than a year. During this time, more popular model-related sensor simulations may have been released on the market.

These two explanations could have both contributed to having a majority of model-related sensor simulations/tools in the sample. Regardless of the exact reasons why, finding that a majority of my sample came from the model-related category was good news. A model-related sensor simulation/tool brings more benefit to the sensor Data Model than a non model-related simulation because it is based on the actual physical properties that exist in the real world. In other words, the value of the sample improves as the number of model-related sensor simulations/tools increases. For that reason, it was beneficial that the research was based on a sample where the majority was model-related. This fact suggests that the final sensor Data Model product is more likely to be relevant even as time passes, computational capacity improves, and model-related sensor simulations become the standard. Additionally, since the sample of sensor simulations/tools represents the diversity of the population of sensor simulations in the

broad M&S community, it is expected that the sensor-related environmental properties that were found during the research will accurately represent the sensor-related environmental properties in the population today. The next section discusses the results of the mapping document development process and describes how the physical properties from the sampled sensor simulations/tools map into SEDRIS.

4.4 Mapping Document Development Process

Research question 3 prompted the exploration of mapping the common sensor input requirements into SEDRIS. During the study of the sensor simulations/tools, the sensor input requirements that SEDRIS needed to capture were determined. These input requirements are referred to as sensor-related environmental properties. The purpose of the mapping document is to match the properties from the sampled sensor simulations/tools to SEDRIS attributes. This process is meant to identify where the differences exist between the sampled sensor simulations/tools and the current Data Model.

To develop the mapping document, all of the sensor simulations/tools' capabilities and characteristics were listed and the sensor-related environmental properties were grouped where a physical commonality existed. In many cases, although some simulations or tools give different names to their capabilities and characteristics, the physical properties they are based on was the same. For instance, one simulation may term one of their capabilities as "far shore brightening" while another simulation may measure "reflection from the sun." Those names are simply non-technical descriptions of

the physical property of reflectivity. Hence, the sensor-related environmental properties were grouped where a physical commonality existed.

For each unique sensor-related environmental property, a determination was made if there was an existing counterpart in the current Data Model. In a few instances matches were found under the Data Table class structure (previously shown in Figure 4.2.1). For example, the sensors in the CCTT simulator use the physical property of "Maximum Temperature" from the SE database. The Data Dictionary already has SE_DATA_ TABLE_LABEL_ENUM of type definition SE_DT_TEMPERATURE_ MAXIMUM. These are put into perspective using Figure 4.2.1. A <Geometry> has zero or more <Data Tables References> which is associated with exactly one <Data Table>, which has one or more ordered <Signature Elements>. One of the value_labels (or SE_DATA_TABLE_ LABEL_ENUMs) in this example is maximum temperature. So the CCTT property of maximum temperature had already been represented in the current SEDRIS Data Model.

In most instances, however, there was no counterpart in the Data Model. In those cases, a corresponding SEDRIS attribute was created. The SEDRIS attributes were defined using the appropriate references, provided by Long Nguyen, including such technical sources such as the IR Handbook, the Modeling & Simulation Scene Generation Attributes Standard, the Photonics Dictionary, etc. This ensured that the mapping document would be understandable without having to continually seek the reference materials. Additionally, a properly written mapping document is the template used to write the SEDRIS Change Requests for modifying the Data Model and updating

the Data Dictionary. The last step was to write the actual mapping document. Figure 4.4.1 is an excerpt from the mapping that shows three of the SEDRIS attributes. The full document is included in Appendix D.

SEDRIS Attribute and Description	Sampled Sensor Simulation/Tool	Sampled Sensor Simulation/Tool Term(s)
Reflectivity: Ratio of reflected (specularly, diffusely, or otherwise) flux divided by incident flux. (unitless)	RADSIM™	- Modifiable reflectivity values - Leading edge enhancement - Far shore brightening
	SensorVision™	- Reflection from sun
	Camber Radar Toolkit™	- Far shore brightening
	SOF ATS	- Reflectivity
	TTS BDD3	- Reflectivity
	IRGen®	- Solar reflectivity
Emissivity: Ratio of the emission of a sample to that of an ideal blackbody at the same temperature and in the same spectral interval. (unitless)	SensorVision™	- Thermal emission
	CCTT	- Emissivity
	Irma	- Thermal emissions - Angle-dependent surface emission
	TTS BDD3	- Emissivity (directivity)
	PRA STM Maps	- Infrared radiance
Transmissivity: Ratio of transmitted flux to incident flux per kilometer; note that this includes direct as well as scattered transmission and therefore is not necessarily related to transmission loss; i.e., transmissivity plus transmission_loss does not necessarily equal unity. (1/km)	IRGen®	- Long-IR emissivity
	CCTT	- Atmospheric transmittance
	RADSIM™	- Cultural feature shadowing - Terrain masking and terrain following
	Irma	- Path transmittance effects - Shadowing
	SOF ATS	- Transparency
	TTS BDD3	- 3D feature shadowing - Alpha (transparency)
	IRGen®	- Cloud transmission

Figure 4.4.1. Excerpt from SEDRIS to Sensor Simulation/Tool Mapping Document

The mapping document has three columns. The first column lists the SEDRIS attribute and its technical definition or description. The second and third column provides the name of the sampled sensor simulation/tool and the term(s) that it uses to describe the SEDRIS attribute, respectively. The mapping document further chronicles the results of the research concerning the common physical properties that need to be present in the SE during runtime so that the sensor simulations/tools can provide useful output to the simulation.

4.5 Results of the Mapping Document Analysis

After completing the mapping document, an analysis was conducted. As Figure 4.4.1 shows, the second and third columns of the mapping document show how many simulations/tools from the sample had sensor-related environmental properties that could be grouped together to describe one SEDRIS attribute. In some cases, six or more simulations/tools required a particular physical property, such as reflectivity or emissivity, to function properly. In other cases, as can be seen in Appendix D, only one simulation/tool from my sample used a certain physical property. When that was the case, that particular physical property was usually one that was simulation-specific to that sensor simulation.

An example of a simulation-specific property is the sensor-related environmental property used by the sensors in the CCTT called Support Temperature Codes. Only the CCTT simulation uses these seven enumerated codes. They indicate how the temperature of a particular material (or object) is being effected by its immediate environmental

surroundings. For instance, Support Temperature Code 5 is the Warm Temperature support code. It should be used for supporting structures that get warm by trapping heat in attics or are near heat sources. The typical uses of code 5 would be roofs of buildings and surfaces somewhat near (but not directly around) heat sources such as engines.

The mapping document development process (from section 4.4) identified *where* the gap existed between the sampled sensor simulations/tools and the current SEDRIS Data Model. The analysis of the mapping document determined *what* was required to resolve the differences. As previously stated, in a few instances the sensor property in the sample matched existing attributes in the Data Model. Since the SEDRIS attribute already contained the sensor property (in those few cases), there was no action required. However in the majority of the cases, a discrepancy *did* exist. The general findings and the necessary corrective actions are summarized in Table 4.5.1.

Table 4.5.1

General Findings from Mapping Document Analysis

General Finding from Analysis	Corrective Action Required
1. Missing an enumeration type that is generally an axis in a data table.	1. Create a new enumeration type.
2. Missing the type of data table that is required to describe the physical property.	2. Create a new data table type enumeration.
3. Missing the type of value that tells you *what* is being represented by a cell value in a data table.	3. Create a new data table label enumeration.
4. Missing the unit designation for the type of value in a data table cell.	4. Create a new unit enumeration.
5. There is no discrepancy found.	5. No action is required.

Using the five general corrective actions listed in Table 4.5.1 as a guide to remedy the discrepancies that were discovered during the analysis of the mapping document, a detailed list of specific "actions required" was prepared with the help of Long Nguyen. This list contains the information that the SEDRIS Team needed in order to modify the Data Model. Additionally, through the research of related sensor attribute documents and discussions with sensor and SEDRIS subject matter experts, some other sensor-related aspects that were needed in the Data Model for proper functioning were determined. Table 4.5.2 presents the combined results.

Table 4.5.2

Specific Findings and Actions Required to Modify the SEDRIS Data Model

Specific Finding from Mapping Document Analysis	Action Required to Modify the Data Model
1. Missing a polarization enumeration	1. Create a polarization enumeration
2. Missing a radar significant factor (RSF) enumeration	2. Create a RSF enumeration
3. Missing a frequency band enumeration for EM emissions	3. Create an EM band enumeration
4. Missing a data table type for absorptivity	4. Create a data table type enumeration for absorptivity
5. Missing a data table type for backscatter	5. Create a data table type enumeration for backscatter
6. Missing a data table type for contrast	6. Create a data table type enumeration for contrast
7. Missing a data table type for emissivity	7. Create a data table type enumeration for emissivity

Specific Finding from Mapping Document Analysis	Action Required to Modify the Data Model
8. Missing a data table type for irradiance	8. Create a data table type enumeration for irradiance
9. Missing a data table type for radar cross section (RCS)	9. Create a data table type enumeration for RCS
10. Missing a data table type for radiance	10. Create a data table type enumeration for radiance
11. Missing a data table type for reflectivity	11. Create a data table type enumeration for reflectivity
12. Missing a data table type for refraction	12. Create a data table type enumeration for refraction
13. Missing a data table type for sun shading	13. Create a data table type enumeration for sun shading
14. Missing a data table type for surface roughness	14. Create a data table type enumeration for surface roughness
15. Missing a data table type for secondary textures	15. Create a data table type enumeration for secondary textures
16. Missing a data table type for transmissivity	16. Create a data table type enumeration for transmissivity
17. Missing a data table type for thermophysical properties	17. Create a data table type enumeration for thermophysical properties
18. Missing a data table label for absorptivity	18. Create a new data table label enumeration for absorptivity
19. Missing a data table label for direct downwelling	19. Create a new data table label enumeration for direct downwelling
20. Missing a data table label for ground clutter	20. Create a new data table label enumeration for ground clutter
21. Missing a data table label for sea clutter	21. Create a new data table label enumeration for sea clutter

Specific Finding from Mapping Document Analysis	Action Required to Modify the Data Model
22. Missing a data table label for the coefficient of convection	22. Create a new data table label enumeration for the coefficient of convection
23. Missing a data table label for diurnal depth	23. Create a new data table label enumeration for diurnal depth
24. Missing a data table label for bi-directional reflectance function (BRDF)	24. Create a new data table label enumeration for BRDF
25. Missing a data table label for diffuse reflectivity	25. Create a new data table label enumeration for diffuse reflectivity
26. Missing a data table label for specular reflectivity	26. Create a new data table label enumeration for specular reflectivity
27. Missing a data table label for reflectivity	27. Create a new data table label enumeration for reflectivity
28. Missing a data table label for the EM band	28. Create a new data table label enumeration for the EM band
29. Missing a data table label for fine scale correlation	29. Create a new data table label enumeration for fine scale correlation
30. Missing a data table label for large scale correlation	30. Create a new data table label enumeration for large scale correlation
31. Missing a data table label for fine scale roughness	31. Create a new data table label enumeration for fine scale roughness
32. Missing a data table label for large scale roughness	32. Create a new data table label enumeration for large scale roughness
33. Missing a data table label for the imaginary refractive index	33. Create a new data table label enumeration for the imaginary refractive index
34. Missing a data table label for the real refractive index	34. Create a new data table label enumeration for the real refractive index

Specific Finding from Mapping Document Analysis	Action Required to Modify the Data Model
35. Missing a data table label for the incident azimuth	35. Create a new data table label enumeration for the incident azimuth
36. Missing a data table label for the incident elevation	36. Create a new data table label enumeration for the incident elevation
37. Missing a data table label for interior temperature	37. Create a new data table label enumeration for interior temperature
38. Missing a data table label for interior flow velocity	38. Create a new data table label enumeration for interior flow velocity
39. Missing a data table label for light level	39. Create a new data table label enumeration for light level
40. Missing a data table label for polarization	40. Create a new data table label enumeration for polarization
41. Missing a data table label for radar cross section (RCS)	41. Create a new data table label enumeration for RCS
42. Missing a data table label for radar significant factor (RSF)	42. Create a new data table label enumeration for RSF
43. Missing a data table label for radiance	43. Create a new data table label enumeration for radiance
44. Missing a data table label for radiance amplitude	44. Create a new data table label enumeration for radiance amplitude
45. Missing a data table label for radiance azimuth	45. Create a new data table label enumeration for radiance azimuth
46. Missing a data table label for radiance elevation	46. Create a new data table label enumeration for radiance elevation
47. Missing a data table label for radiance phase	47. Create a new data table label enumeration for radiance phase
48. Missing a data table label for relative radiance	48. Create a new data table label enumeration for relative radiance

Specific Finding from Mapping Document Analysis	Action Required to Modify the Data Model
49. Missing a data table label for diffused lunar radiance	49. Create a new data table label enumeration for diffused lunar radiance
50. Missing a data table label for diffused solar radiance	50. Create a new data table label enumeration for diffused solar radiance
51. Missing a data table label for direct lunar radiance	51. Create a new data table label enumeration for direct lunar radiance
52. Missing a data table label for direct solar radiance	52. Create a new data table label enumeration for direct solar radiance
53. Missing a data table label for reflected azimuth	53. Create a new data table label enumeration for reflected azimuth
54. Missing a data table label for reflected elevation	54. Create a new data table label enumeration for reflected elevation
55. Missing a data table label for secondary texture	55. Create a new data table label enumeration for secondary texture
56. Missing a data table label for solar absorptivity	56. Create a new data table label enumeration for solar absorptivity
57. Missing a data table label for specific heat	57. Create a new data table label enumeration for specific heat
58. Missing a data table label for sun shading	58. Create a new data table label enumeration for sun shading
59. Missing a data table label for support temperature code	59. Create a new data table label enumeration for support temp code
60. Missing a data table label for surface backscatter	60. Create a new data table label enumeration for surface backscatter
61. Missing a data table label for volume backscatter	61. Create a new data table label enumeration for volume backscatter
62. Missing a data table label for thermal conductivity	62. Create a new data table label enumeration for thermal conductivity

Specific Finding from Mapping Document Analysis	Action Required to Modify the Data Model
63. Missing a data table label for thermal contrast	63. Create a new data table label enumeration for thermal contrast
64. Missing a data table label for visual contrast	64. Create a new data table label enumeration for visual contrast
65. Missing a data table label for thickness	65. Create a new data table label enumeration for thickness
66. Missing a data table label for total transmissivity	66. Create a new data table label enumeration for total transmissivity
67. Missing a data table label for transmissivity	67. Create a new data table label enumeration for transmissivity
68. Missing a data table label for transmission attenuation	68. Create a new data table label enumeration for transmission attenuation
69. Missing a data table label for transmission loss	69. Create a new data table label enumeration for transmission loss
70. Missing a data table label for transmitted azimuth	70. Create a new data table label enumeration for transmitted azimuth
71. Missing a data table label for transmitted elevation	71. Create a new data table label enumeration for transmitted elevation
72. Missing a data table label for wavelength	72. Create a new data table label enumeration for wavelength
73. Missing many unit designations for the types of values in the data table cells.	73. Create many new unit enumerations. See Appendix E for SCR defining units
74. The data table label for maximum temperature already exists in SEDRIS	74. No action required
75. The data table label for density already exists in SEDRIS	75. No action required
76. The data table label for thermal mass already exists in SEDRIS	76. No action required

The results presented in Table 4.5.1 and Table 4.5.2 show recommendations for the creation of 3 new enumeration types, 14 new data table type enumerations, 55 new data table label enumerations, and 1 new units enumeration that defines many sensor-related unit designations. Additionally, there were 3 instances where no action was required because the attribute that was needed already existed in the Data Model. Presenting these recommendations concludes the mapping document analysis and identifies the input requirements that need to be represented in SEDRIS.

4.6 Results of the Modification of the SEDRIS Data Model

The results shown in Table 4.5.2 were submitted to the SEDRIS Team for peer review, conversion into SEDRIS Change Requests (SCR), and inclusion in the Data Model. Although many SEDRIS Team members aided in this process, the review and authoring of the SCRs was completed primarily by Dr. Paul Berner then of Analysis and Technology, Inc. (A&T). Once the SCRs were generated, they were sent to Michele Worley at SAIC. She meticulously reviewed the SCRs to ensure that they adhered to the proper formatting standards and followed all SEDRIS rules before coding the changes into the Data Model.

4.6.1 Explanation of Sensor-Related SEDRIS Change Requests

The changes recommended in Table 4.5.2 were captured by Dr. Berner in two documents, SCR# A&T-017 and SCR# A&T-019. The first SCR, A&T-017, defines the unit enumeration, called SE_UNIT_ENUM. The type definitions for SE_UNIT_ENUM

contain units of measurement for the types of values in a Cell of a Data Table. The type definitions include units of measurement for time, length, area, speed, mass, force, work, power, density, temperature, and many more. The units that were needed to represent values for sensor representation were only a part of this SCR. Dr. Berner wrote this SCR to make changes for the SEDRIS associates concerned with units used in the Data Table class. Therefore, SCR# A&T-017 represents the combination of the results of the mapping document analysis with his own findings and the discrepancies found by many others on the SEDRIS Team concerning units of measurement. Appendix E shows a text copy of SCR# A&T-017.

The second SCR, A&T-019, adds new data table types and data value enumerations that are needed to support EM and other physical properties of materials in the SE. This SCR, which implemented the bulk of the sensor portion of the Data Model, was the result of my recommendations in Table 4.5.2 for creating 3 new enumeration types, 14 new data table type enumerations, and 55 new data table label enumerations. A text copy of SCR# A&T-019 is shown at Appendix F. For the most part, A&T-019 lists the enumerations that are in Table 4.5.2. The 3 new enumeration types, however, also list the type definitions. As Appendix F illustrates, the 3 new enumeration types are polarization, RSF, and EM band, called SE_DT_POLARIZATION_ENUM, SE_DT_RADAR_SIGNIFICANT_FACTOR_ ENUM, and SE_DT_EM_BAND_ ENUM, respectively. The type definitions for these enumerations list the 10 types of polarization, the 14 types of radar significant factors, and the 26 different EM band types that were recommended and later reviewed by Long Nguyen.

To illustrate, SCR# A&T-019 lists the type definitions for SE_DT_ POLARIZATION_ENUM as follows:

```
typedef enum
{
    SE_POLARIZATION_ALL,             /* All or any
    SE_POLARIZATION_RANDOM,          /* Random */
    SE_POLARIZATION_CIRCULAR,        /* Circular */
    SE_POLARIZATION_ELLIPTICAL,      /* Elliptical */
    SE_POLARIZATION_HH,              /* Horizontal */
    SE_POLARIZATION_VV,              /* Vertical) */
    SE_POLARIZATION_HV,              /* Crossed */
    SE_POLARIZATION_VH,              /* Crossed */
    SE_POLARIZATION_S,               /* S (perpendicular to incidence-reflectance
                                     plane) */
    SE_POLARIZATION_P                /* P (parallel to incidence-reflectance
                                     plane)*/
} SE_DT_POLARIZATION_ENUM;
```

This enumeration signifies the type of polarization in EM or light radiation. Polarization is generally an axis in an EM data table type. According to Dr. Berner's example in SCR# A&T-019, a SE_TABLE_EM_REFLECTIVITY table type may have SE_DT_BRDF as one of its labels and SE_DT_POLARIZATION_ENUM as one of its axes. This data table would indicate that BRDF is a function of EM polarization, i.e., reflectance can take on different values for random, circular, crossed, or any other enumerated type of polarization listed in the type definition above.

4.6.2 Implementation of the SEDRIS Data Coding Standard

As the data model grew in size and complexity, the number of additions of new data table type enumerations and new data table label enumerations used in the Data

Table class became overbearing. More importantly, some SEDRIS team members were concerned that as the use of SEDRIS by the M&S user community gained in popularity, the frequency that users would require these types of additions would increase at a faster rate than the semi-annual Data Model update releases. For these reasons, the SEDRIS management decided to make two inter-related changes.

First, they decided to create the SEDRIS Data Coding Standard (SDCS). According to Dr. Paul Birkel (1999), the SDCS unifies characterizations of both individual primitives and structured collection of environmental "things." The assortment of codes that make up the SDCS answers three types of questions about environmental things: (1) what is it; (2) what are its additional clarifying characteristics; and (3) how does it deviate from normality. The first portion of the SDCS that answers question (1) is called the SEDRIS Classification Codes (SCC). These 5-character codes are used to *classify* environmental things. The codes that pertain to question (2) are called SEDRIS Attribute Codes (SAC). These 4-character codes further describe the environmental things by their *attributes*. The last segment of the SDCS that deals with question (3) are called SEDRIS State Codes (SSC). These 4-character codes provide information about an environmental thing's current *state*, if different than normal.

An example will clarify the purpose of and relationship between the SCC, SAC, and SSC. Suppose the environmental thing we want to fully describe is a 3-meter wide bridge that is 50 percent damaged. In SDCS descriptive terms: the SCC tells you what it is – classified as a bridge; the SAC tells you the bridge's clarifying characteristics – has an attribute of three meters wide; and the SSC tells you how the bridge deviates from its

normal state – it is 50 percent damaged. The actual codes, names, and descriptions for the bridge example using the SDCS version 2.0 are shown in Table 4.6.2.

Table 4.6.2

Example of Bridge Using the SDCS 2.0

Type/Code	Name	Description
SCC – AQ040	Bridge/Overpass/Viaduct	A man-made structure spanning and providing passage over a body of water, depression, or other obstacles.
SAC – WID_	Width	For a bridge, the width is the measurement perpendicular to the axis between the abutments.
SSC – DMAN	Damage, Maneuver	The extent of physical injury/damage to the capability to maneuver, in terms of percent degradation from a fully capable state.

(SEDRIS Data Coding Standard, 1999)

Generally, the SDCS follows the same standardized coding system used in the Feature and Attribute Coding Catalogue (FACC). The FACC is the fourth part of the Digital Geographic Information Exchange Standard (DIGEST), which is the international exchange standard applicable to all member nations of the Digital Geographic Information Working Group (DGIWG). The FACC "provides a means for encoding real world entities or objects for the purpose of an orderly exchange of digital geospatial information between organizations" (Digital Geographic Information Working Group, 1997, Ch. 5). Initially the SEDRIS management intended to simply use the FACC codes as the standardized coding system for use with SEDRIS. However, since the acceptance

timeline for changes to the FACC standards were considered too long and inflexible to meet the SEDRIS user's needs, it was more useful to use a separate standard. Therefore, in addition to leveraging the current DIGEST approach by capturing all the FACC codes, the SDCS was designed with the ability to handle the larger set of environmental characterizations that SEDRIS users require. Dr. Paul A. Birkel of The MITRE Corporation led the main effort for the SDCS development. Dr. Birkel maintains the latest version of the SDCS and manages it as an integrated, multi-table Access® 97 database.

The second inter-related change that the SEDRIS management decided to make was to separate the SDCS from the Data Model. Through separation, the SDCS enumerations and the Data Model can evolve at different rates. This allows the SEDRIS Team to release an updated version of the SDCS database more frequently than the Data Model. This process is similar to the methodology used in the virus protection software industry. The base virus protection software doesn't release a new version every time a new virus is discovered. They simply update the virus definition file. That way, customers can quickly download and install the small file that interfaces seamlessly with the base software already installed on their operating system.

The SDCS is scheduled for updated every two to three months, whereas the Data Model will be updated approximately every six months. It is likely that the only time the Data Model version number and the SDCS version number will ever be the same is when they were both first released as versions 2.0 to the public on January 7, 1999. In general, the separation approach will make SEDRIS more useful and less expensive to maintain in

terms of time and money. Specifically, for all users interested in sensor simulations or tools, the creation of the SDCS makes SEDRIS's extensibility more standardized and timely.

The SDCS is of particular interest to this research because the timing of its implementation affected the inclusion of SCR# A&T-019 into the Data Model. Recall that A&T-019 represented all of the recommendations to modify the Data Model with respect to sensor-related environmental attributes. As the SDCS was developed, the SCCs were intended to replace the data table type enumerations (SE_DATA_TABLE_TYPE_ENUM) from the Data Model, while the SACs were intended to replace the data table label enumerations (SE_DATA_TABLE_LABEL_ENUM). However, in order to meet the same January 7, 1999, public release deadline scheduled for the Data Model, the SDCS 2.0 had to be released before it was fully populated.

By the time this research entered the demonstration phase, the SDCS 2.0 had not yet captured the recommended data table type enumerations from SCR# A&T-019 as SCCs. In fact, the SDCS 2.0 did not include *any* of the SCCs required to use Data Tables. The candidate list of SDCS enumerations was complete, but Dr. Birkel simply did not have enough time to include them all and still meet the release deadline. The SDCS 2.0 did, however, include all of the recommended sensor-related data table label enumerations as SACs from SCR# A&T-019. When the entire SAC listing was thoroughly inspected, the findings indicated that in addition to capturing the recommended data table label enumerations as SACs, Dr. Birkel had created several complementary sensor-related SACs to ensure consistency with other entries in the

listing. The SAC listing in the SDCS 2.0 contains approximately 1400 items. To save space, only the sensor-related SACs are shown in Appendix G.

In the months following the demonstration, the SDCS was updated to include the appropriate SCCs for sensor-related data table types. The original set of data table types recommended in SCR# A&T-019 (from Table 4.5.2) were as follows:

SE_TABLE_EM_ABSORPTIVITY,
SE_TABLE_EM_BACKSCATTER,
SE_TABLE_EM_CONTRAST,
SE_TABLE_EM_EMISSIVITY,
SE_TABLE_EM_IRRADIANCE,
SE_TABLE_EM_RADAR_CROSS_SECTION,
SE_TABLE_EM_RADIANCE,
SE_TABLE_EM_REFLECTIVITY,
SE_TABLE_EM_REFRACTION,
SE_TABLE_EM_SUN_SHADING,
SE_TABLE_EM_SURFACE_ROUGHNESS,
SE_TABLE_EM_SECONDARY_TEXTURE,
SE_TABLE_EM_TRANSMISSIVITY,
SE_TABLE_THERMOPHYSICAL

In an effort to make SEDRIS more user-friendly to consumers, Long Nguyen suggested that these 14 data table types should be further generalized. After further analysis, five new data table types, now called SCCs, were nominated. They were included in SCR # PDB-007 that was submitted on March 1, 1999, by Dr. Paul Berner. The following list is an excerpt from SCR # PDB-007 showing the five new SCCs scheduled for inclusion in the SDCS 2.1 release.

- SE_SCC_MATERIAL_CHARACTERISTICS
  Physical properties of material(s).
  <Data Table> support.

- SE_SCC_ELECTROMAGNETIC_CHARACTERISTICS
  Electromagnetic properties of material(s).
  <Data Table> support.

- SE_SCC_RADAR_CHARACTERISTICS
  Radar properties of material(s).
  <Data Table> support.

- SE_SCC_INFRARED_CHARACTERISTICS
  Infrared properties of material(s).
  <Data Table> support.

- SE_SCC_THERMAL_CHARACTERISTICS
  Thermal properties of material or systems(s).
  <Data Table> support.

This approach allowed the addition of capability without the addition of too much complexity. A fewer number of sensor-related data table types means that consumers can locate the data they are interested in more easily with less confusion.

Although the peer review, SCR authoring, and SDCS development will continue throughout the SEDRIS life cycle, the release of the 2.0 versions of the Data Model and SDCS signaled the end of the implementation phase of this research. At this point the modifications, that would make the conduct of the minimal database interchange experiment possible, were complete.

4.7 Results of the Minimal Database Interchange Experiment

This section discusses the details of the conduct of the minimal database interchange experiment, the results obtained, and the analysis of those results. As previously stated, the experiment was conducted with the 2.0 versions of the Data Model and SDCS, which were the most current versions available at the time. Unfortunately, based on the research timeline, waiting until the SDCS 2.0 was fully complete before

conducting the test was not an option. As a result, some SCCs that were needed were not available for use during the experiment.

4.7.1 Conduct of the Interchange Experiment

The minimal database interchange experiment was conducted as previously described in chapter 3. It may be helpful to periodically refer back to Figure 3.8.2. Russ Moulton, Jr., of JRM Enterprises, Inc., provided a test database from the "Paint-the-Night" (PTN) IR sensor simulation. The test database consisted of a one-polygon PTN data set with its associated sensor-related properties. He built the sample PTN data set so that it contained eight sensor-related environmental attributes on each of the three vertices of the polygon. Although it is a small SE sensor database, this minimal database represents data in the PTN database format complete with eight of the same attributes that are contained in the full PTN database.

As Table 4.7.1 shows, the sample data set, called database *A*, is composed of three vertices. Each vertex is represented in the synthetic environment at a 3-D location, *x*, *y*, and *z* (in meters) within the Universal Transverse Mercator (UTM) Projected Coordinate System. Additionally, each vertex has a normal vector array, which describes the local orientation at that point, referred to in the PTN data set as *i, j*, and *k*. Lastly, there are eight sensor-related properties associated with each vertex. Appendix H shows the minimal PTN database computer code, including the header file and the corresponding data values for grid location, normal vector location, and the sensor properties.

Table 4.7.1

Description of Database *A*

Value Label	Vertex 0	Vertex 1	Vertex 2
x ; positive Eastward	0	1	0
y ; positive Northward	0	0	1
z ; elevation	0	0	0
i ; normal vector	0	0.577	-0.577
j ; normal vector	0	-0.577	0.577
k ; normal vector	1	0.577	0.577
eL ; emissivity, longwave	0.92	0.85	0.8
elwir ; emissivity, long-IR	0.8	0.8	0.8
emwir ; emissivity, mid-IR	0.5	0.5	0.5
aS ; absorptivity, solar	0.2	0.2	0.2
h ; coefficient of convection	4.0	4.0	4.0
kt ; thermal conductivity	0.52	0.062	0.062
p ; density	1840	920	920
ch ; specific heat	1500	1104	1104

Using the first step in the two-step procedure to replicate the Producer Write API process that was described in chapter 3, the *description* of the PTN database (from Table 4.7.1 and Appendix H) was mapped to the 2.0 version of the Data Model. Naturally, only the classes that were needed to adequately capture the PTN database were used. Figure 4.7.1 shows those SEDRIS classes and their relationships using Data Model notation.

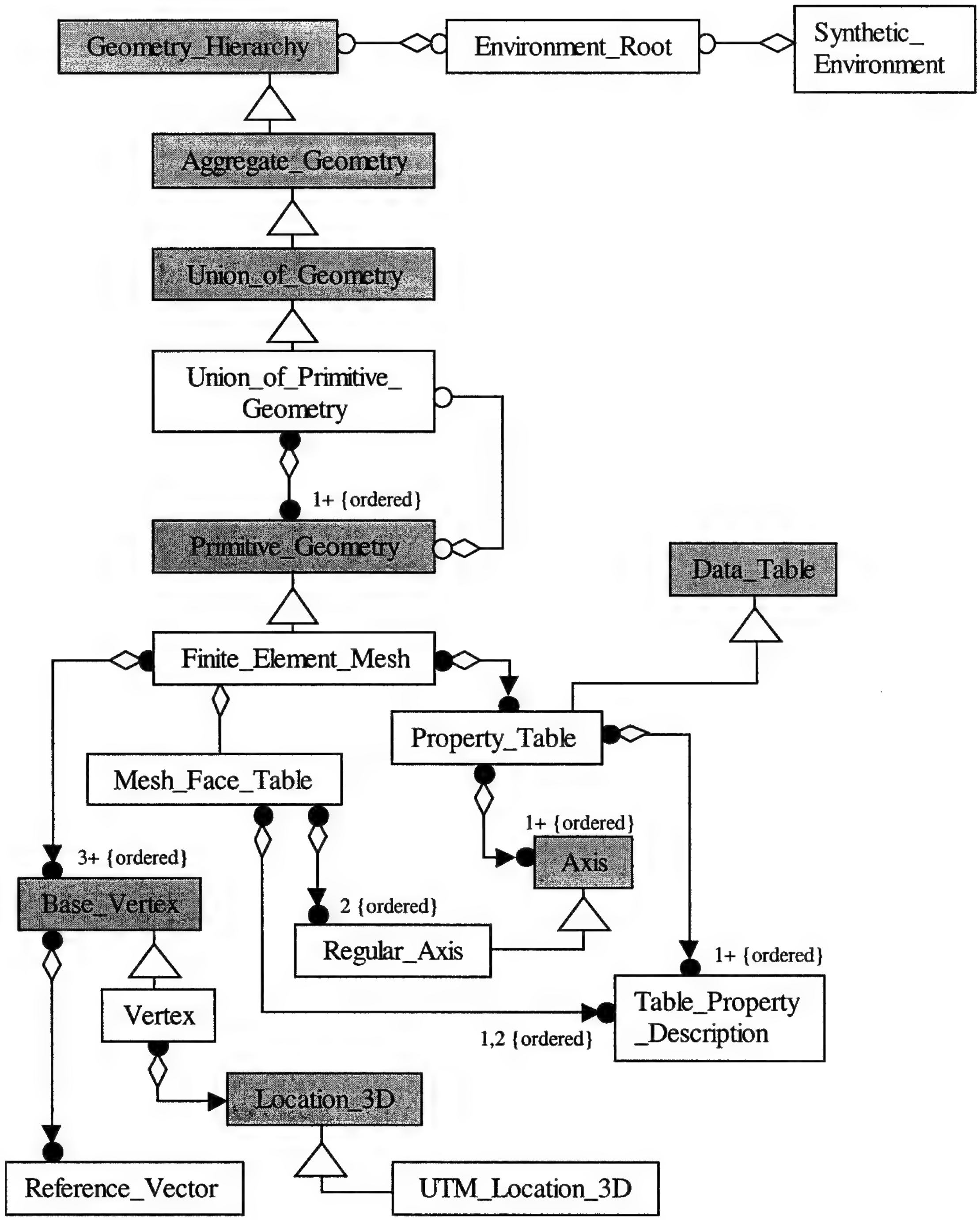


Figure 4.7.1. Mapping of the Minimal PTN Database to SEDRIS Data Model Classes

Next, using the generalized SEDRIS class diagram in Figure 4.7.1 as a starting point, a detailed object diagram was created. The PTN minimal interchange experiment object diagram at Appendix I is a representation of the PTN data in terms of instances of corresponding Data Model classes complete with the legal entries from the Data Dictionary in the required fields. The second step of the two-step procedure was for Bill Horan of SAIC to create a SEDRIS transmittal using the object diagram as the source document. He constructed the SEDRIS database transmittal using a special software utility he designed that performs an in-memory database generation function. After constructing the transmittal, the transmittal was validated and viewed using the SEDRIS tools.

Using Checker, a plain text program, the syntactical correctness of all entries in the test transmittal was validated in accordance with the Data Model. Another useful tool we used was Depth, which provides a plain text printout of the entire transmittal. During creation of the transmittal, Bill Horan used Depth periodically to debug data relationships. When the test transmittal was completed, Depth was used to create a text file of the PTN test transmittal for comparison and presentation purposes. In its "very verbose" mode, Depth prints out the entire contents of a transmittal including the data values. Appendix J shows a copy of the PTN test interchange experiment transmittal printed in Depth's very verbose mode. At this point, both the minimal native database file in PTN format, called *A*, and a STF database file based on a description of the minimal PTN native database were ready for data comparison.

The last step in the demonstration was to convert the SEDRIS data in STF into the target database format, which happened to be the same as the PTN native database format, using a Consumer Read API. Due to a protracted delay in funding, JRM Enterprises, Inc., was not funded in time to write the Consumer Read API code that was needed to consume the STF data. Fortunately, SAIC was able to commit additional resources to the research endeavor. Bill Horan, playing the role of the sensor data consumer, wrote the necessary code that consumed the STF data, converted it into the target PTN database format and produced the output shown in Appendix K. The resulting database, called A', represents a consumer's differing runtime format. Table 4.7.2 displays the output data (from Appendix K) in a tabular format.

Table 4.7.2

Description of Database A'

Value Label	Vertex 0	Vertex 1	Vertex 2
x ; positive Eastward	0.000000	1.000000	0.000000
y ; positive Northward	0.000000	0.000000	1.000000
z ; elevation	0.000000	0.000000	0.000000
i ; normal vector	0.000000	0.577000	-0.577000
j ; normal vector	0.000000	-0.577000	0.577000
k ; normal vector	1.000000	0.577000	0.577000
eL ; emissivity, longwave	0.920000	0.850000	0.800000
$elwir$; emissivity, long-IR	0.800000	0.800000	0.800000
$emwir$; emissivity, mid-IR	0.500000	0.500000	0.500000

Value Label	Vertex 0	Vertex 1	Vertex 2
aS ; absorptivity, solar	0.200000	0.200000	0.200000
h ; coefficient of convection	4.000000	4.000000	4.000000
kt ; thermal conductivity	0.520000	0.062000	0.062000
p ; density	1840.000000	920.000000	920.000000
ch ; specific heat	1500.000000	1104.000000	1104.000000

4.7.2 Analysis of the Interchange Experiment

The analysis of the minimal database interchange experiment was conducted according to the two-stage data comparison procedures detailed in section 3.8.3. The first stage data comparison took place after database A was converted into STF data. The second stage data comparison took place after the STF data was converted into A' data. In both stages, the databases were compared to determine whether or not the data interchange was loss-less and accurate. Recall that a *loss-less data interchange* is considered a success if no data is lost or is determined to be missing upon data comparison and an *accurate data interchange* is considered a success if no data is changed in any way during the data interchange process, including rounding errors.

4.7.2.1 First Stage Data Comparison

In the first stage data comparison, A compared to STF, the loss-less goal was for every data value in A is to have a unique storage location in STF. A careful inspection of

the appendices containing the initial PTN database (*A*), the object diagram, and the resulting transmittal (STF) shows that the interchange *was* loss-less. The native database structures and data values were preserved in STF. SEDRIS did not add or remove anything from the original native database. Although it was able to capture all the data in a loss-less manner, there were some important findings/issues that warrant discussion.

The first finding relates to the Table Property Descriptions class. As Figure 4.7.1 shows, a Property Table (which is a type of Data Table) is composed of (among other things) one or more ordered Table Property Descriptions. The Table Property Description components of a Property Table describe how to interpret the corresponding property values retrieved from the cells of a Property Table. One of its required field elements is attribute_code, which is a 4-character SAC. Earlier, the analysis of the mapping document (Appendix D) resulted in a list of recommended additions to the Data Model. Those additions were first captured in SCR# A&T-019 (Appendix F) and then as SACs in the SDCS 2.0 (Appendix G). One of those SACs was for emissivity. It has a 4-character code of EMS_ and a label of SE_SAC_EMISSIVITY. When the object diagram was created, the findings indicated that the emissivity SAC that was recommended was inadequate to describe the minimal PTN database's sensor-related properties.

In database *A* (shown in Table 4.7.1), there are three different types of emissivity attributes on each vertex: longwave emissivity, long-IR emissivity, and mid-IR emissivity. To successfully describe the PTN database, a SAC was needed for each of the three types of emissivity attributes. The EMS_ code was only adequate for longwave

emissivity. Therefore, SACs for both long-IR emissivity and mid-IR emissivity had to be created to successfully demonstrate the model. These are the two SACs that were created to support the demonstration:

- EMSA Emissivity, Long Infrared SE_SAC_EMISSIVITY_LONG_IR
- EMSB Emissivity, Mid Infrared SE_SAC_EMISSIVITY_MID_IR

The next finding is very similar. Each of the PTN database vertices has a density attribute. Although the SDCS 2.0 contained SACs for many different kinds of density measures, it did not have a SAC for just plain density. Before the creation of the SDCS, the Data Model *did* contain a plain density enumeration, so it either fell through the cracks during the transition or was better described by several SACs. In either case, a plain density SAC would eventually have to be created to support a full PTN database interchange. To support the minimal interchange experiment, a substitute SAC for density called Air Density, Climatology – Mean (ADCM) was used. It was similar in definition and had the same units of was measurement – kilograms per meter cubed.

The last issue relates to the SDCS 2.0 being incomplete at the time of the demonstration. As previously stated, the native PTN database consists of three vertices, which is the fewest number of vertices that define a polygon. The easiest method to map the PTN database into SEDRIS would have been to use the Polygon class, which is a Primitive Geometry. As Figure 4.7.1 shows, the Polygon class was not used. Instead, the Finite Element Mesh class was used, which also is a Primitive Geometry. This decision was made because the full PTN database will need to use the Finite Element Mesh class when an ideal interchange experiment is conducted. According to the SEDRIS Data

Dictionary (1999), "a Finite Element Mesh is a tesselation [an arrangement of a pattern of small polygons] of a surface into mesh faces [usually triangles]. There may be data associated with each vertex and/or mesh face. Knowing which vertices form a mesh face is important for interpolation and other computations."

As Figure 4.7.1 indicates, a Finite Element Mesh is composed of three or more ordered Base Vertices, one or more Property Tables, and exactly one Mesh Face Table. Both the Property Table and the Mesh Face Table classes (which are types of Data Tables) require the field element data_table_type. Recall from section 4.6.2 that data table types are identified by a 5-character SCC. Since the SDCS 2.0 release was not fully populated at the time of my demonstration, the 5-character SCCs that was needed did not exist.

The SCC labels, however, *were* available. The labels and their detailed descriptions had been included in the Data Model as enumerations prior to converting to the SDCS methodology, but they were subsequently removed to avoid duplication. Therefore, the object diagram (Appendix I) contains the correct labels for the Property Table and the Mesh Face Table data_table_types, but the SCC shown in the text printout of the transmittal (Appendix J) uses a fictitious 5-character SCC. For Property Table, the label is SE_SCC_MESH_VERTEX_PROPERTIES and for Mesh Face Table, the label is SE_SCC_MESH_FACE_TABLE. Since the goal was to develop and demonstrate the sensor portion of the Data Model, it is immaterial whether or not the correct 5-character SCC existed when the test was conducted. Therefore, the imaginary code of XX999 was used in both data_table_type field elements. To ensure that the SCCs for the support of

Data Tables was included in the SDCS 2.1 release, Dr. Berner included them in SCR # PDB-007 that was submitted on March 1, 1999.

As part of the first stage data comparison, it was also checked whether or not the data interchange was accurate. The accuracy goal was for every data value in A to have the exact same value as in STF. A comparison of the data values in the native database (Appendix H) and the SEDRIS transmittal (Appendix J) indicated that the interchange *was* accurate. The data values were not changed in any way during the data interchange process, including rounding errors.

4.7.2.2 Second Stage Data Comparison

The second stage data comparison, which took place after the STF data was converted into A' data, compared A data to A' data. In the second data comparison, the loss-less goal was to have the same number and type of data values in both A and A'. A comparison of the data values (presented in both Appendices H & K and Tables 4.7.1 & 4.7.2) showed that the interchange *was* loss-less. The A database has 14 data values that are numerical and the A' database has 14 data values that are numerical.

The accuracy goal was for every data value in A to have the exact same value as in A'. A comparison of the data values in the native database and the target database (presented in both Appendices H & K and Tables 4.7.1 & 4.7.2) indicated that the interchange *was* accurate. The data values were not changed in any way during the data interchange process, including rounding errors.

In summary, the demonstration was successful. The minimal sensor database interchange experiment was loss-less and accurate in both stages of the data comparison. In chapter 5, some general conclusions are drawn and the additional Data Model recommendations are presented.

CHAPTER 5

CONCLUSIONS, LESSONS LEARNED AND EXTENSIONS TO THE RESEARCH

5.1 Conclusions

The primary goal of this research effort was to examine and extend the current *capability* of the SEDRIS Data Model to fully include sensor representation. Recall from chapter 2 that the research centered around the question: How can the SEDRIS Data Model be extended to include sensor representation? To answer the main research question and accomplish the primary goal, five subordinate research questions were examined.

5.1.1 Summary of Research Questions

The first research question was: Why did the SEDRIS Team decide that using the Data Table class was the most efficient and unambiguous method for SEDRIS to support sensor simulation data using the current (1.04d) SEDRIS Data Model structure? The research indicated that the Data Table class gives the database producers and consumers the most flexibility for fully representing the SE compared to the available alternatives.

Research question 2 was: What sensor input requirements need to be represented in SEDRIS? By studying a sample of sensor simulations/tools that represent the diversity within the military-oriented sensor simulation community, the research yielded the

sensor-related environmental properties (input requirements) that SEDRIS needed to capture in the Data Model.

The third research question was: Can these common sensor simulation input requirements be mapped into SEDRIS? It was determined that, yes, the common input requirements *could* be mapped into SEDRIS. To prove it, a mapping document was created that mapped each sensor-related environmental property to a SEDRIS attribute. When the appropriate attribute did not exist in SEDRIS, one was defined.

Research question 4 was: Can a list of "actions required" be identified that will be useable by the SEDRIS Team to modify the Data Model to include the desired sensor input requirements? The analysis of the mapping document showed that, yes, an "actions required" list could be developed. As each new SEDRIS attribute during the mapping document development process was defined, it was placed it on the list of "actions required." This list of recommended changes was subsequently used to create the SCRs that modified the Data Model to include sensor representation (and were later used during the SDCS development). The final result was the SEDRIS public release of the 2.0 versions of the Data Model and SDCS.

The last research question was: Can a minimal sensor database interchange experiment using a modified Data Model be conducted to demonstrate if the data interchange was both loss-less and accurate? The minimal database interchange experiment using a small sensor database from the PTN IR sensor simulation demonstrated that the interchange of sensor data could be both loss-less and accurate.

As the summary of the five research sub-questions shows, this research achieved its primary goal. Through this research, the *capability* of the SEDRIS Data Model was able to be extended to fully include sensor representation. Note the emphasis is placed on the word *capability*. This work has extended the capability of SEDRIS to fully include sensor representation. This contributes to the body of knowledge within the SEDRIS community both a conceptual framework for sensor representation as well as the concrete technical additions necessary to interchange sensor databases.

Unfortunately, this research does not extend SEDRIS to *fully* include sensor representation. That degree of representation may never be achieved. It is more likely that each sensor simulation/tool that is mapped into SEDRIS will require a slight addition or modification to the Data Model or SDCS. The minimal database interchange experiment is an excellent example. Although the demonstration was successful, in that both stages of the data comparison proved the interchange was loss-less and accurate, it also highlighted that the SDCS portion of the modified Data Model needed some improvements for full functionality.

5.1.2 Extensibility

The minimal database interchange experiment proved exactly what it was intended to – that the database comparisons of both A to STF and A to A' were a success in terms of the loss-less and accuracy goals. However, it also revealed a few deficiencies in terms of the emissivity and density SACs. This finding suggests that the SAC listing is incomplete regarding sensor-related environmental attributes. It also lends credibility

to the notion that although the *capability* of SEDRIS has been extended to fully include sensor representation, it has not been *fully* extended SEDRIS to include all sensor representation.

The demonstration also illuminates the one of the greatest by-products that this research contributed to the SEDRIS project – extensibility. Not only does this research provide the conceptual framework and concrete technical additions for sensor representation, it also provides the capability to easily and quickly extend the sensor portion of the Data Model in the future.

The research indicated that the most difficult aspect of using SEDRIS is learning to "speak" the SEDRIS language. It requires getting familiar with the Data Model, the Data Dictionary, the SDCS, the Geospatial Reference Model, and the SEDRIS tools and utilities, just to name a few. Once comfortable with SEDRIS, this research will help any new sensor simulation producer or consumer become "SEDRIS compliant" quickly and easily. Due to the introduction of the SDCS and the sensor-related additions from this research, the new SEDRIS user will likely find that the only changes that are needed to interchange their sensor data using SEDRIS are some minor additions to the SACs. These additions will likely only account for the model-specific elements in their simulation that were not general enough to discover in this research, such as the Support Temperature Code element used by the CCTT sensors that was described earlier. In these cases, the process to extend SEDRIS to meet the user's needs will be easy – simply recommend/nominate the appropriate SAC using a SCR. It will also be quick – most likely under two months since the SDCS evolves at a faster rate than the Data Model.

Another reason this research adds extensibility to SEDRIS is because it recommended the inclusion of some environmental attributes in SEDRIS that currently are not being modeled in sensor simulations, but may be modeled in the future. Examples of these types of environmental attributes are the real index of refraction and the imaginary index of refraction from the Modeling & Simulation Scene Generation Attributes Standard (Cornette, 1996, p. 19). The SACs and labels in the SDCS 2.0 are:

- RIR_ SE_SAC_REFRACTIVE_INDEX_REAL_PART
- RII_ SE_SAC_REFRACTIVE_INDEX_IMAGINARY_PART

A more detailed explanation of these attributes can be found in Appendix G. A sensor simulation using these attributes in the SE would be on the high side of the fidelity continuum. It is likely that as computer processors become less expensive and more computationally robust, some members of the M&S community will use these types of attributes in their sensor simulation models. Because of the foresight to recommend some future environmental attributes that may be needed, the benefits of this research are very likely to be pertinent even as time passes.

Probably the most important contribution that this research has provided is the development of a process that can be followed by other researchers and database developers. As mentioned earlier in this chapter, by extended the capability of SEDRIS to fully include sensor representation, it has contributed a conceptual framework for sensor representation within the SEDRIS community. As intended, this research will primarily benefit members of the sensor simulation community who need to use SEDRIS as an interchange format. However, by researching and implementing a method for

extending the SEDRIS Data Model, this thesis developed a step-by-step process that can be generalized and applied to other application areas (other than sensors) in the SEDRIS user community. By documenting a proven, effective process, it is also intended to serve as a template that newcomers to SEDRIS can follow to reduce the time it takes to understand and extend SEDRIS to suite their individual needs. So that others may benefit from its development, the step-by-step process is summarized below.

1. Study the SEDRIS Data Model to determine if the existing Data Model structure can accommodate the specific application area of interest.
2. Research the population of simulation applications in the area of interest.
3. Select a representative sample by focusing on recommendations from SMEs and popular simulations in the application area.
4. Study the sample to determine what requirements from the application area of interest need to be represented in SEDRIS.
5. Map the requirements into SEDRIS using the Data Model, Data Dictionary, and SDCS. Identify where the gaps exist between SEDRIS and the sample simulation applications.
6. Analyze the gaps. Create, group, and list the needed attributes.
7. Write the SCRs (or nominate the findings to the SME) to make the appropriate SEDRIS modifications.
8. Follow up to ensure that the modifications were made.
9. Test the modified Data Model using a database interchange experiment.
10. Analyze the results and make the appropriate adjustments.

5.1.3 Recommended Additions to the SDCS

As alluded to in chapter 4, several more SACs were recommended for inclusion in SDCS 2.1 subsequent to the demonstration. When the minimal PTN interchange experiment showed that the SDCS was incomplete, further analysis of sensor-related SACs was conducted. This is consistent with step 10 of the process that was just described. In addition to the emissivity and density attributes that PTN required, several SACs were nominated to better define the environment with respect to sensors. Table 5.1.3 shows the complete list of the SAC additions that were recommended post-demonstration.

Table 5.1.3

Recommended SAC Additions

SAC	Name	Label	Units	Data Type
DNSY	Density	SE_SAC_DENSITY	KG/M^3	FLOAT 32
EMSA	Emissivity, Long Infrared	SE_SAC_EMISSIVITY_ LONG_IR	UNITLESS	FLOAT 32
EMSB	Emissivity, Mid Infrared	SE_SAC_EMISSIVITY_ MID_IR	UNITLESS	FLOAT 32
EMSC	Emissivity, Near Infrared	SE_SAC_EMISSIVITY_ NEAR_IR	UNITLESS	FLOAT 32
EMSL	Emissivity, Longwave	SE_SAC_EMISSIVITY_ LONGWAVE	UNITLESS	FLOAT 32
SREF	Surface Reflectivity, Far Infrared	SE_SAC_SURFACE_ REFLECTIVITY_FAR_IR	UNITLESS	FLOAT 32

SAC	Name	Label	Units	Data Type
SREM	Surface Reflectivity, Mid Infrared	SE_SAC_SURFACE_ REFLECTIVITY_MID_IR	UNITLESS	FLOAT 32
SREN	Surface Reflectivity, Near Infrared	SE_SAC_SURFACE_ REFLECTIVITY_NEAR_IR	UNITLESS	FLOAT 32
SREV	Surface Reflectivity, Visible	SE_SAC_SURFACE_ REFLECTIVITY_VISIBLE	UNITLESS	FLOAT 32

5.2 Lessons Learned

An undertaking as complex as extending SEDRIS for sensor representation seemed to be quite a daunting task. In the end, the findings indicated that the SEDRIS Project was very close to its first objective, capturing the complete set of environmental data elements and their relationships in a data model, before the research began. From a Data Model developer's point-of-view, the first priority has to be capturing the relationships between data elements. Then, the focus can shift to capturing the *complete* set of data elements from the diverse M&S community. The SEDRIS Team already had great success towards the first priority and was sharply focused on the second priority when this research started. Had this not been the case, it is doubtful that the research efforts would have been successful.

In the beginning, the complexity of SEDRIS was overwhelming. After climbing the learning curve, as all newcomers to SEDRIS must do, the findings indicate that the Data Model was very flexible and well-suited to extension. The experience during the

minimal database interchange experiment confirmed what many database producer's had said before – that the toughest part of working with SEDRIS to produce an STF database was the data mapping process. As the object diagram was created for the simplistic native PTN database, it was amazing to see the number of choices that were available to represent the same data. This finding falls directly in line with SEDRIS Guiding Axiom 5, "the Data Model should use as many standard definitions for data elements as possible" (STRICOM, 1998b, p. 4). Although these choices made the task somewhat difficult, it also allowed the achievement of an unambiguous representation of the data elements while providing a great deal of flexibility.

The documentation step, while usually the least enjoyable, becomes the most important aspect of any software-related endeavor after its released to the public. The SEDRIS Team has done a superb job throughout the development of SEDRIS of staying on top of documentation. It is usually very difficult to find any authoritive information about a new topic such as SEDRIS. The SEDRIS Team's approach towards documentation facilitated a quick determination of what prior work had been accomplished that was pertinent to the research questions. The SEDRIS technical information web site contains varied sources of information that are able to be both directly viewed and downloaded. Since the details concerning the different SEDRIS components changed so rapidly during the development process, it was difficult to determine which version of the documentation was the most current. Sometimes the viewable version and the downloadable version of the same document did not match. This situation will likely stabilize as SEDRIS continues to mature. As SEDRIS gains

public popularity and more users investigate the SEDRIS documents at the web site, the number of inconsistencies should decrease.

One of the potential drawbacks associated with research that involves a simulation, test, or experiment is the need for outside support. The support necessary for this research included sensor technical support and computer coding technical support – both of which require funding. The sensor support that was needed related to knowledge about sensor simulations and the underlying physics that drive the models. Long Nguyen of NAWC-TSD, as part of the SEDRIS Team, provided unlimited access to his expert knowledge. The computer coding support that was need related to three areas of the demonstration: obtaining the native database, producing the STF data, and consuming the STF data. As previously noted, Russ Moulton of JRM Enterprises, Inc., provided the native database and Bill Horan of SAIC provided the coding support. Since JRM Enterprises was a new SEDRIS associate, the funding stream from DMSO was not fully established in time for my demonstration as previously promised. Each area that requires outside support or funding is an area where the researcher gives up control over the process and, hence, the timeline. The funding problems that were encountered stressed the research timeline and removed all flexibility.

5.3 Further Research

Further exploration of sensor representation in SEDRIS should provide greater insight into defining a Data Model that supports the *full* range of simulation applications. Increasing the sample size of sensor simulations/tools used to derive input requirements

for inclusion in SEDRIS may improve the likelihood that SEDRIS will meet its goal of representational completeness with respect to sensors. To reduce the funding required for a larger sample study, the researcher could capitalize on the future popularity of SEDRIS as the required government interchange specification. As new sensor simulation users naturally enter the SEDRIS market, the required data can be collected after each new user completes the database mapping process. This approach would be very cost-effective, but it would also require a larger timeline for completion.

5.3.1 Nested Data Tables

The findings suggest that subsequent research should follow an incremental approach. The first step should be an experiment with only slight variation of the same minimal database interchange experiment using the same PTN data set. The variation should be in how the sensor-related property values are captured in the Property Table class. The purpose of this demonstration would be to test the Data Model's functionality concerning the use of nested data tables. Nested data tables are Data Tables that refer to other Data Tables. The cell data elements in the primary Data Table contain an index value that points to another Data Table. Although this mapping methodology is more complex, it represents how a data producer can use SEDRIS to represent data in a more efficient and powerful manner

For example, the minimal PTN database contained three vertices, each containing eight sensor-related properties. Recall that to capture these properties in the Data Model, a Property Table (which is a Data Table) of type Mesh Vertex Data was used with a

Regular Axis called Index To Mesh Vertex that has three "tick" marks, one for each of the three vertices. At each "tick" mark along the axis, there are eight Table Property Descriptions describing the thermal property cell data elements by a SAC. (If needed, see Appendix I). Using nested data tables is an alternate method to represent the same data.

A nested data table approach to capture the different sensor-related properties at each of the three vertices would use the same Property Table of type Mesh Vertex Data with the same Regular Axis called Index To Mesh Vertex that has three "tick" marks, one for each of the three vertices. However, instead of eight Table Property Descriptions at each "tick" mark along the axis, it would have only one Table Property Description with a SAC of INDX. This SAC, labeled SE_SAC_INDEX, is used to index multiple values in a related Data Table. The eight thermal properties at each vertex could be represented as a separate Data Table of type SE_SCC_THERMAL_CHARACTERISTICS, which describes thermal properties of materials or systems (introduced in chapter 4). Each vertex, in effect, would have an attribute that is its own thermal system.

This particular method is overly complex and rather inefficient for a small database such as the one used in the PTN experiment. When the database is large, however, the efficiency of this method becomes abundantly clear. To illustrate, say the database contains a flat 5K by 5K area that represents a dirt landing strip with grass along both sides. Further assume that for this area the database producer attributed eight thermal properties to all vertices with a material type of dirt and attributed the same eight thermal properties (with different property values of course) to all vertices with a material

type of grass. In effect we have one thermal system for the landing strip and a second thermal system for the grassy area. Imagine how many vertices could be in a 5K by 5K area. Using the PTN demonstration method, every vertex would have eight Table Property Descriptions with associated values – most having the same cell values. Using the nested data table approach, every vertex would have an index value of 1 or 2 that points to either Thermal System 1 or Thermal System 2, which is much less redundant. Had the PTN database been larger, it could have benefited from the more efficient and powerful nested data table technique.

5.3.2 Full Sensor Database

The next incremental step for further research in the sensor area would be to map a full sensor database into the Data Model, then conduct an ideal database interchange experiment using the data comparison methodology that I described in section 3.8.1. The mapping process associated with a full sensor database would test the validity of both the Data Model and the SDCS more strenuously than the minimal PTN experiment. Additionally, since a full sensor database will likely contain structures and data elements common to all synthetic environments, this ideal interchange experiment would add value to the SEDRIS program in several application areas (besides sensors), including CGF applications and visual systems. As mentioned earlier in the chapter, it is expected that this thesis would benefit the user attempting a full sensor database mapping by serving as a solid foundation from which to start. The findings resulting from an ideal database interchange experiment would be an excellent measure of the validity of this thesis.

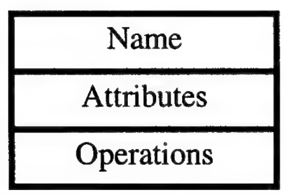
Although it was outside the scope of the data comparison in the minimal PTN demonstration, additional research could compare the size of the database files before and after a full interchange experiment and measure the total time for each conversion to further gauge the efficiency of SEDRIS. In general, further research should follow the mantra, "First see if it can be done, then try to make it more efficient."

Appendix A

SEDRIS Data Model Notation

The SEDRIS data model provides not only a clear description of the data but also defines the relationships between the data – relationships that are critical to ensuring a correct interpretation by its users. This appendix explains the fundamental concepts of modeling representational data types in SEDRIS. The information in this appendix comes directly from James' (1997a) paper titled *SEDRIS Data Model.*

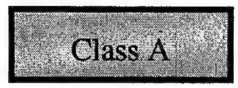
A class of data has a name, a set of attributes that describe its state, and a set of operations that can be performed on the attributes. In OMT notation, a class is represented as a box:



To minimize the amount of information on a diagram of the SEDRIS Data Model, we chose to put all attribute information in the Data Dictionary. Since the data types defined by the SEDRIS Data Model are used to capture the contents of an environmental database, they do not experience dynamic changes. Therefore, no operations are defined for any of the SEDRIS data types. So in the SEDRIS Data Model, a class is represented simply as a rectangle with a Name:

There are two kinds of classes represented in the SEDRIS Data Model: abstract and concrete. Abstract classes never have instantiations; in other words, no objects of

this class type will be created. Concrete classes can have objects created from them. Abstract classes are used to define the common attributes that may be used by any of its subordinate, or child, classes. The relationship between parent and child classes (called an inheritance relationship) will be described more fully in a later paragraph in this section. To denote abstract classes in the SEDRIS Data Model, the class rectangle is shaded:

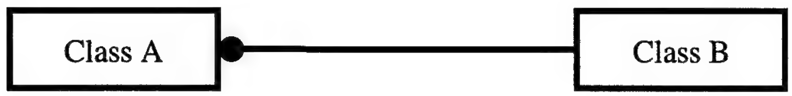


Shading rectangles to denote abstract classes is a deviation from the official OMT notation.

As stated earlier, a critical type of information to be captured in a data model is the relationships between the classes of data. There are three kinds of relationships between classes of data types: association, inheritance, and aggregation. Each relationship type has a different notation. The weakest kind of relationship is association, shown as a simple direct line between two classes:

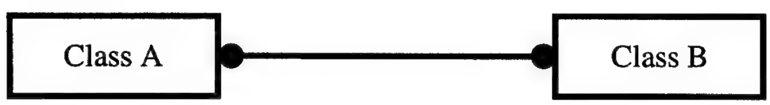
This notation indicates that every Class A is associated with one Class B, and every Class B is associated with one Class A.

Symbols at the end of the relationship lines are used to show multiplicity of the classes. No symbols, as used in the diagram above, indicate exactly one of each class type. A filled circle means zero or *more* classes at that end of the relationship:



This notation says that each Class A class is associated with exactly one Class B class, but each Class B class is associated with zero or more Class A classes. In other words, a Class A class must have an associated Class B class, but a Class B class does not have to have an associated Class A class.

The following diagram says that Class A can have, but is not required to have, an association with Class B; and, Class B may or may not have an association with Class A:

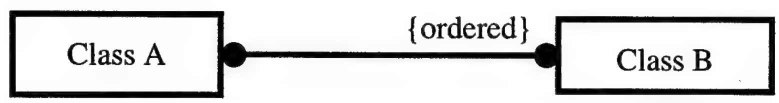


If a number with a plus sign (+) is shown at either end of a relationship line with a closed circle, it means that instead of zero or more classes, there are at *least* the number indicated or more classes. The following diagram illustrates that Class A is associated with at least 3 or more Class B classes. Class B remains associated with zero or more Class A classes.

An open circle at either end of the relationship line indicates zero or *one* class:

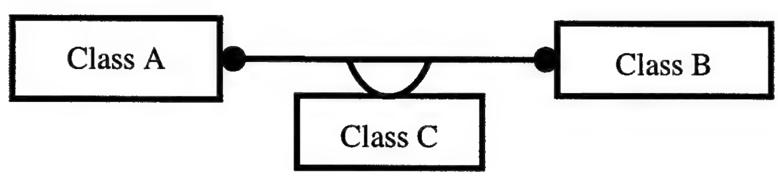
The above notation says that Class A is associated with either none or only one Class B class, while Class B is associated with zero or more Class A classes.

To indicate if the order of the classes is important to the relationship, an {ordered} tag is shown on the appropriate end of the relationship line:



This notation indicates that Class A is associated with zero or many Class B classes and if there are any Class B classes, the order is critical. Unless the end of a relationship line has the {ordered} tag, the collection of classes at that end are considered to be unordered.

The association relationship between two data classes may have its own attributes. To show this, an association class is drawn with a "horse collar" attachment:

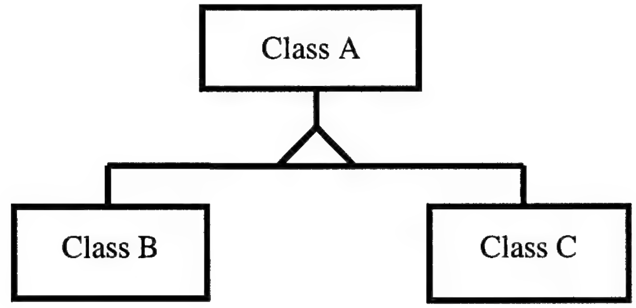


This notation indicates that each Class A is associated with zero or more Class B classes and each Class B is associated with zero or more Class A classes. In addition, it shows

that for any (Class A, Class B) relationship pair, the relationship is also associated with exactly one Class C class.

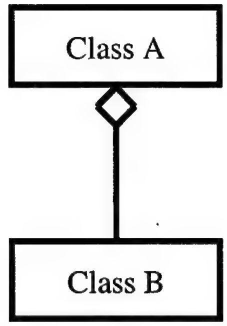
The last notation used with association relationships is the arrowhead to depict one-way associations. This notation is used to indicate that when traversing the data model the relationship can be followed only in the direction of the arrowhead. In the following diagram, Class A is associated with zero or more Class B classes while each Class B is associated with zero or more Class A classes. However, when traversing through the data model, Class A "knows" that it is associated with Class B, while Class B does not "know" that it is associated with Class A. A traversal can go from Class A to Class B, but not from Class B to Class A.

The next type of relationship between two classes is the inheritance relationship, introduced earlier. One of the classes in an inheritance relationship is the Parent or Super Class while the other class (or classes) is the Child or Subordinate Class. In this type of relationship, the Child Classes "inherit" the attributes of their Parent Class, while having unique, additional attributes of their own. This is sometimes referred to as the "is-a" relationship – the Child Class "is-a" type of the Parent Class. The inheritance relationship is depicted in the data model with a triangle:

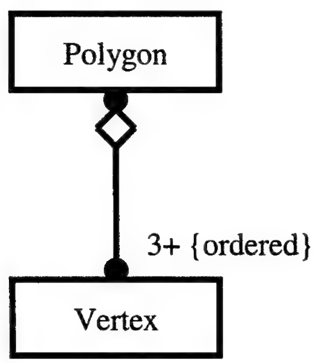


This diagram shows that Class B and Class C are both child classes of Class A. This means that both Classes B and C have the common attributes inherited from Class A and also have unique attributes of there own. If Classes B and C did not have unique attributes that distinguished between them, then they would really be the same class. Sometimes the parent class (in this example, Class A) will be an abstract class (can not be instantiated) while the child classes will be concrete classes (can instantiate or create objects of this type). There is no multiplicity symbols used for the inheritance relationship since the SEDRIS Data Model does not support multiple inheritance. The "is-a" relationship is always one to one. Not allowing multiple inheritance relationships is a deviation from the official OMT notation.

The third and final type of relationship that can be shown between data classes in the SEDRIS Data Model is the aggregation type. This relationship is sometimes referred to as the "has-a" relationship. In this relationship, Class A is an aggregation of (or contains) a Class B class. In other words, Class A "has-a" Class B. This relationship is denoted by a diamond positioned at the aggregate end of the relationship line:

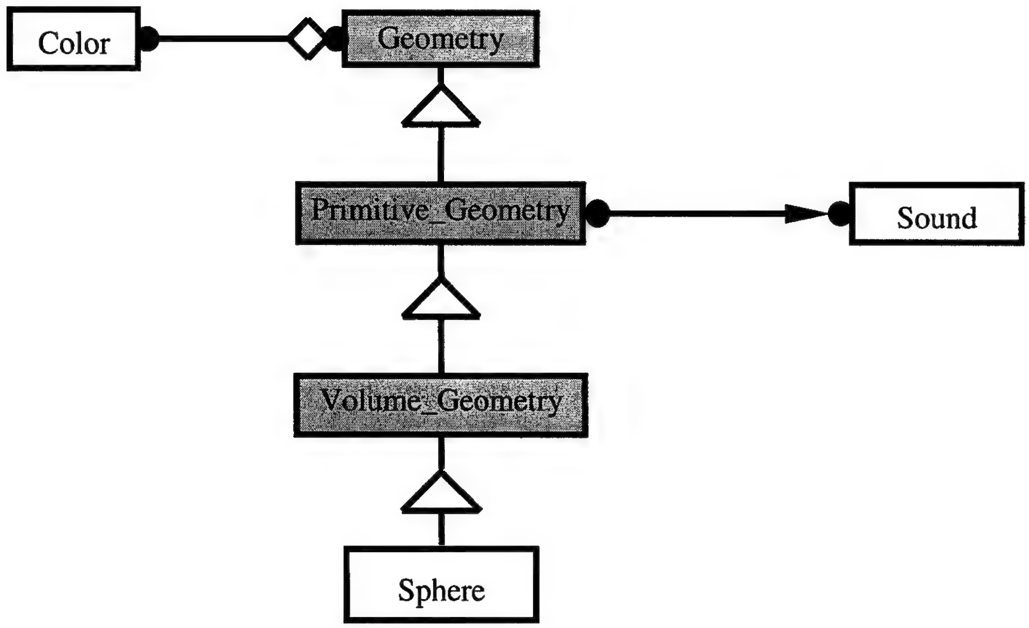


The aggregation relationship can also have multiplicity as described for the association relationship. The same symbology of open and filled circles is used with aggregation relationships:



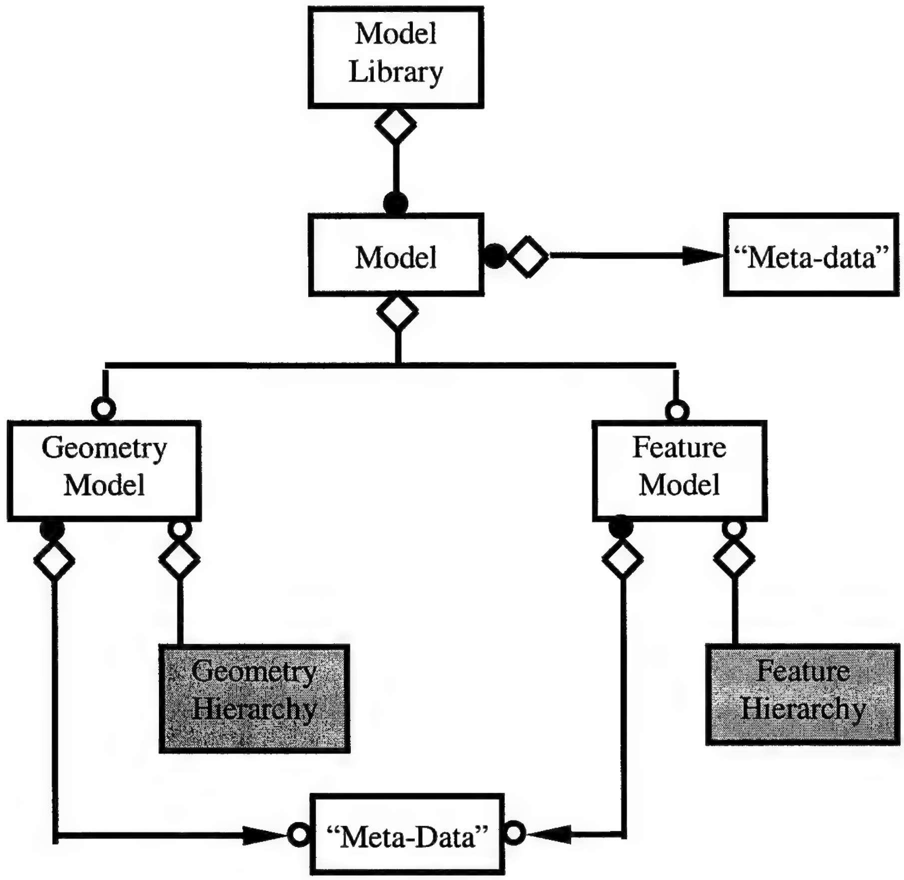
This diagram shows that a Polygon Class is an aggregation of three or more ordered Vertex Classes. A Vertex Class is a component of zero or more Polygon Classes.

The following example shows how a piece of the SEDRIS Data Model would be interpreted:



This notation shows that a Sphere Class is-a Volume_Geometry; is-a Primitive_Geometry; and, is-a Geometry. So, the Sphere Class has inherited attributes from all its Parent Classes as well as has its own unique attributes. In addition, a Sphere Class has zero or more Sound Classes associated with it. The Sphere Class is composed of zero or more Color Classes. The Sphere Class "knows" that it may have a Sound Class, but the Sound Class does not know it is associated with a Sphere Class.

The following illustration shows a level of the Data Model. Although the class names are similar to those in the Data Model, this diagram is not a true representation of all the classes and relationships that would be used to define this portion of the SEDRIS Data Model.

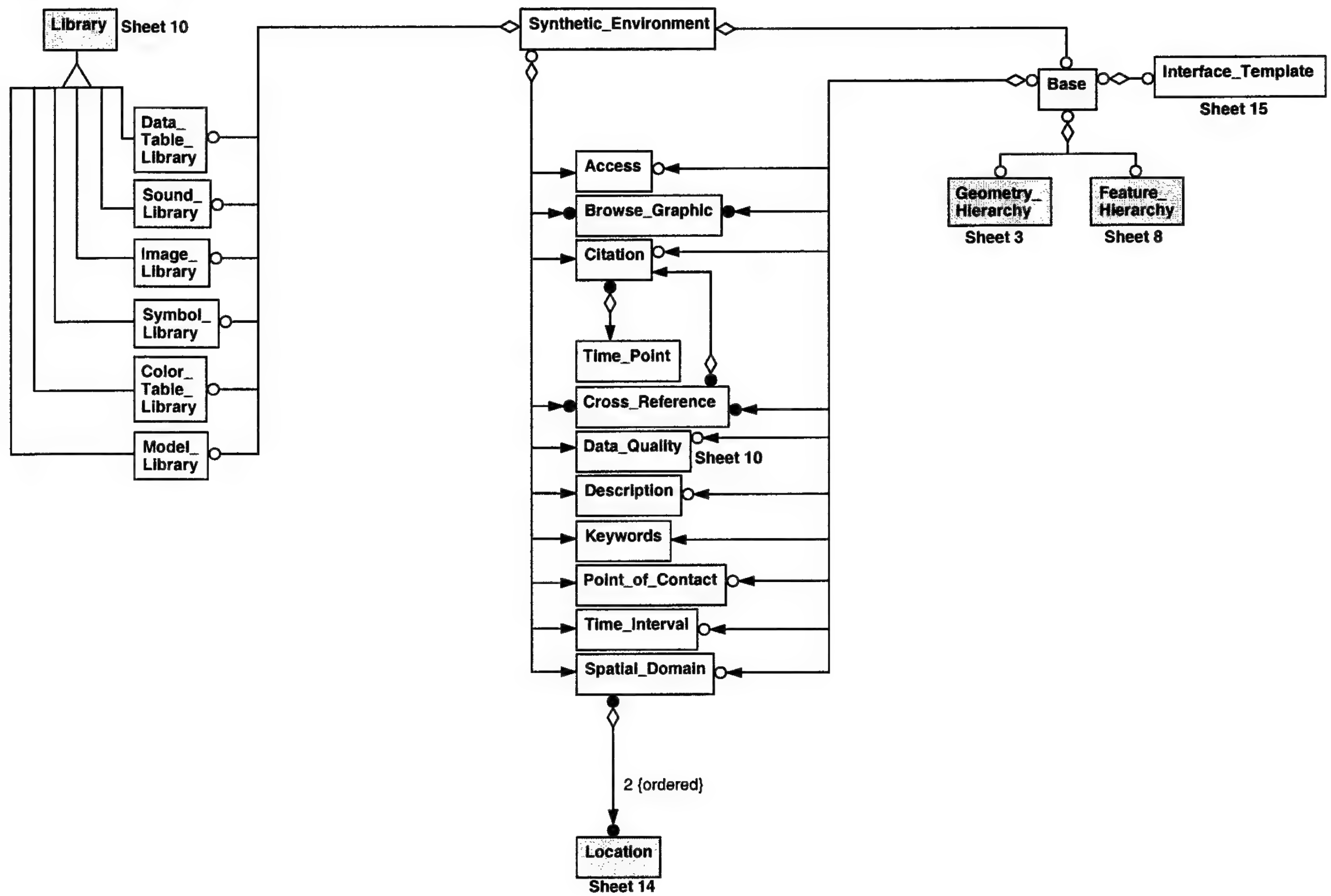


This diagram shows the Model_Library Class is an aggregation of zero or more Model Classes. A Model Class is composed of either a Geometry_Model or a Feature_Model. The Geometry_Model Class is an aggregation of the sub-classes of the Geometry_Hierarchy abstract class. Similarly, the Feature_Model Class is composed of sub-classes of the Feature_Hierarchy abstract class. The Model, Geometry_Model, and Feature_Model Classes each have a set of "Meta-Data" information describing them. Obviously, the Geometry_Hierarchy and Feature_Hierarchy Classes, since they are both abstract classes, have further levels of sub-classes that would be shown on other sheets of the Data Model.

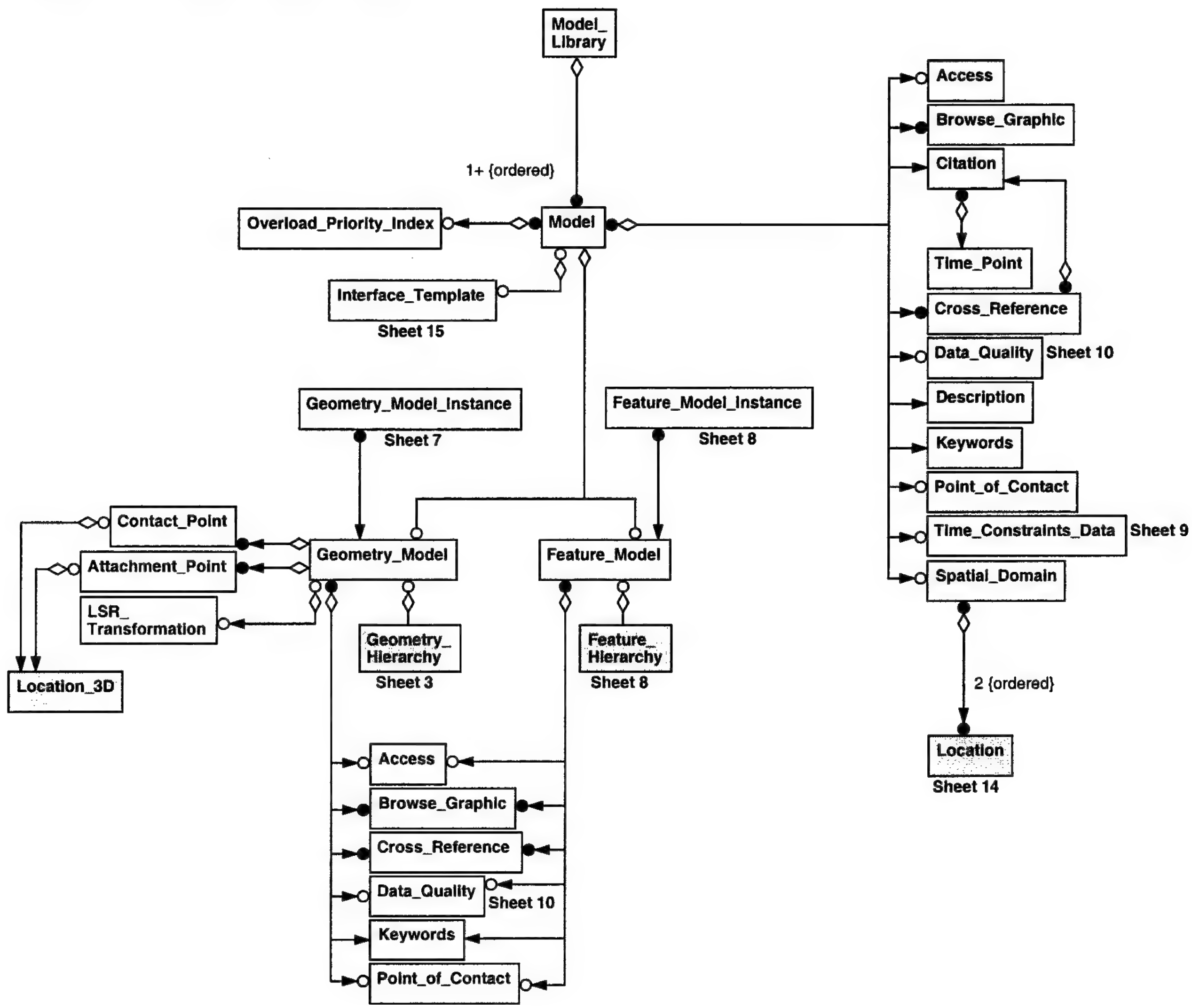
Appendix B

SEDRIS Data Model Version 1.04d

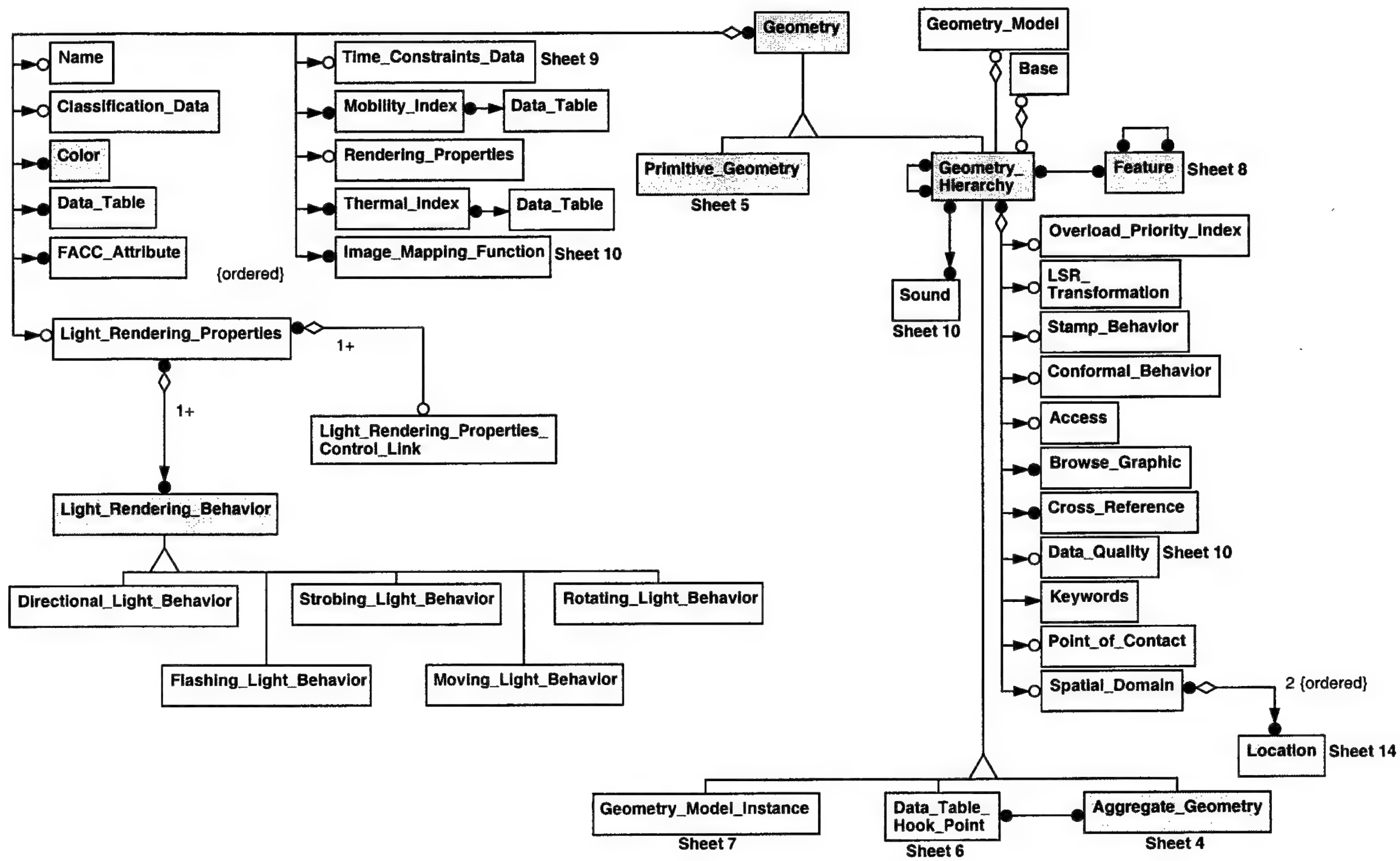
Sheet: 1: Synthetic Environment, World & Model Panel: (0,0)



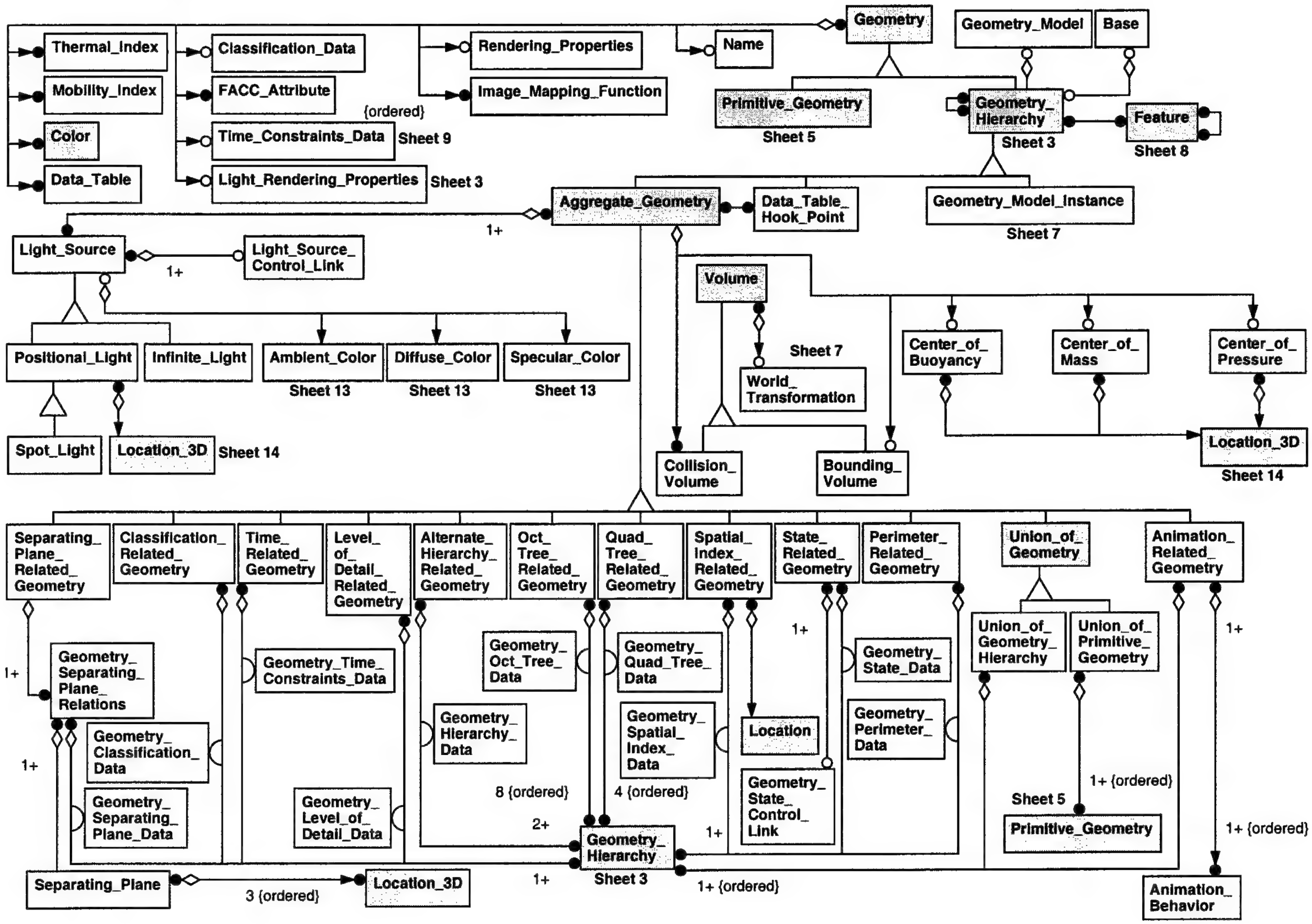
Sheet: 2: Model Panel: (0,0)



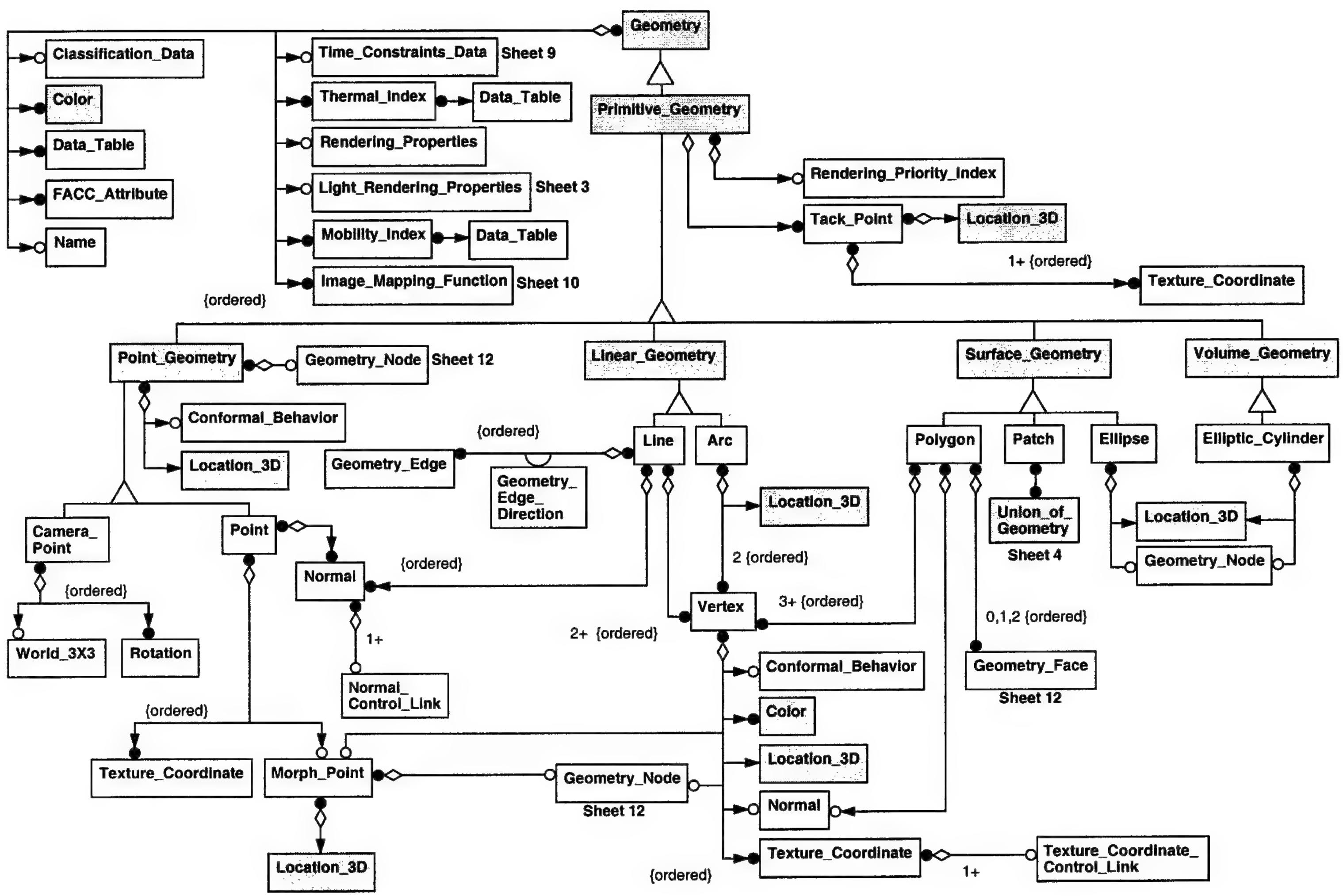
Sheet: 3: Geometry, Geometry Hierarchy, & Light Rendering Properties Panel: (0,0)



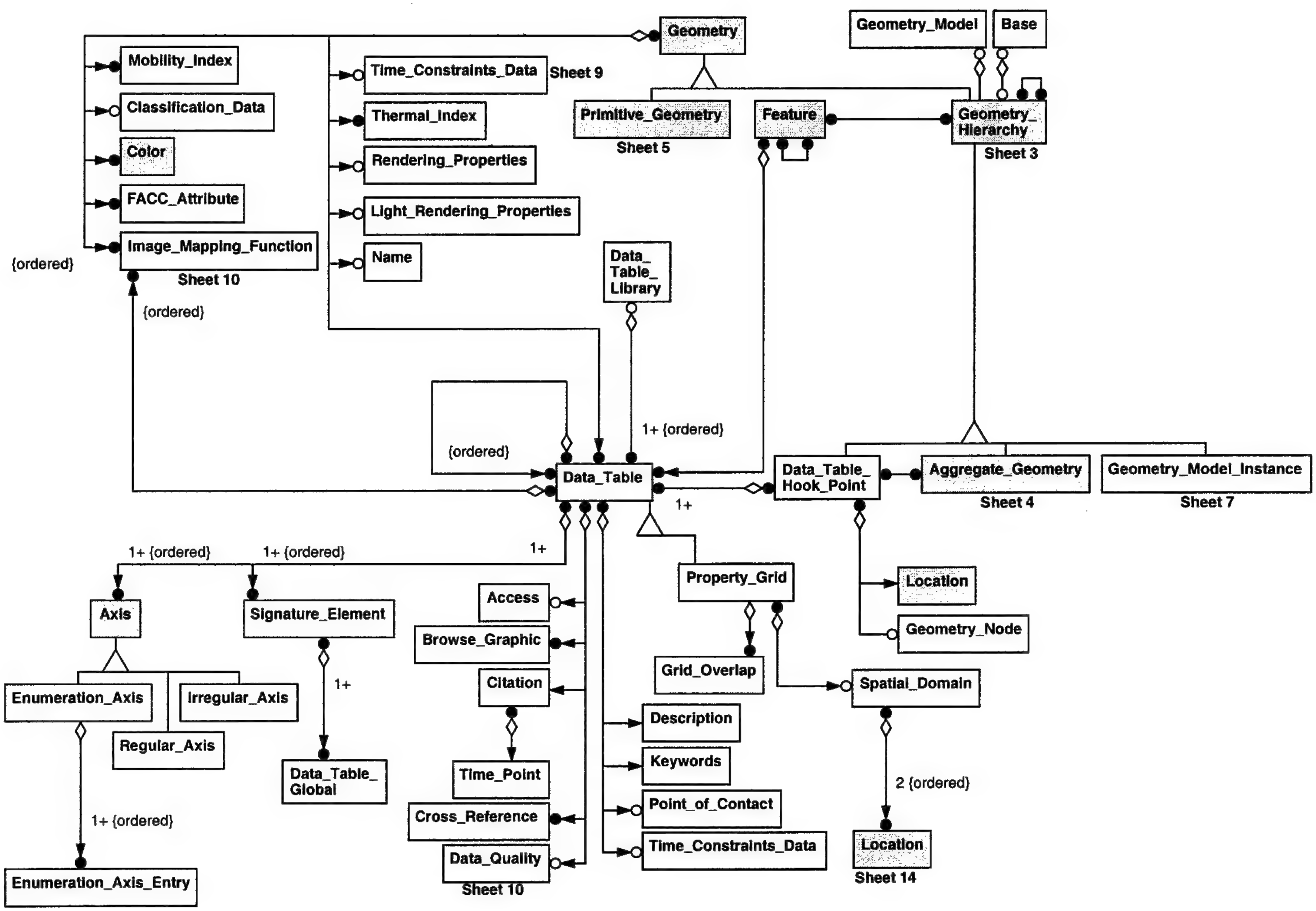
Sheet: 4: Aggregate Geometry & Light Source Panel: (0,0)



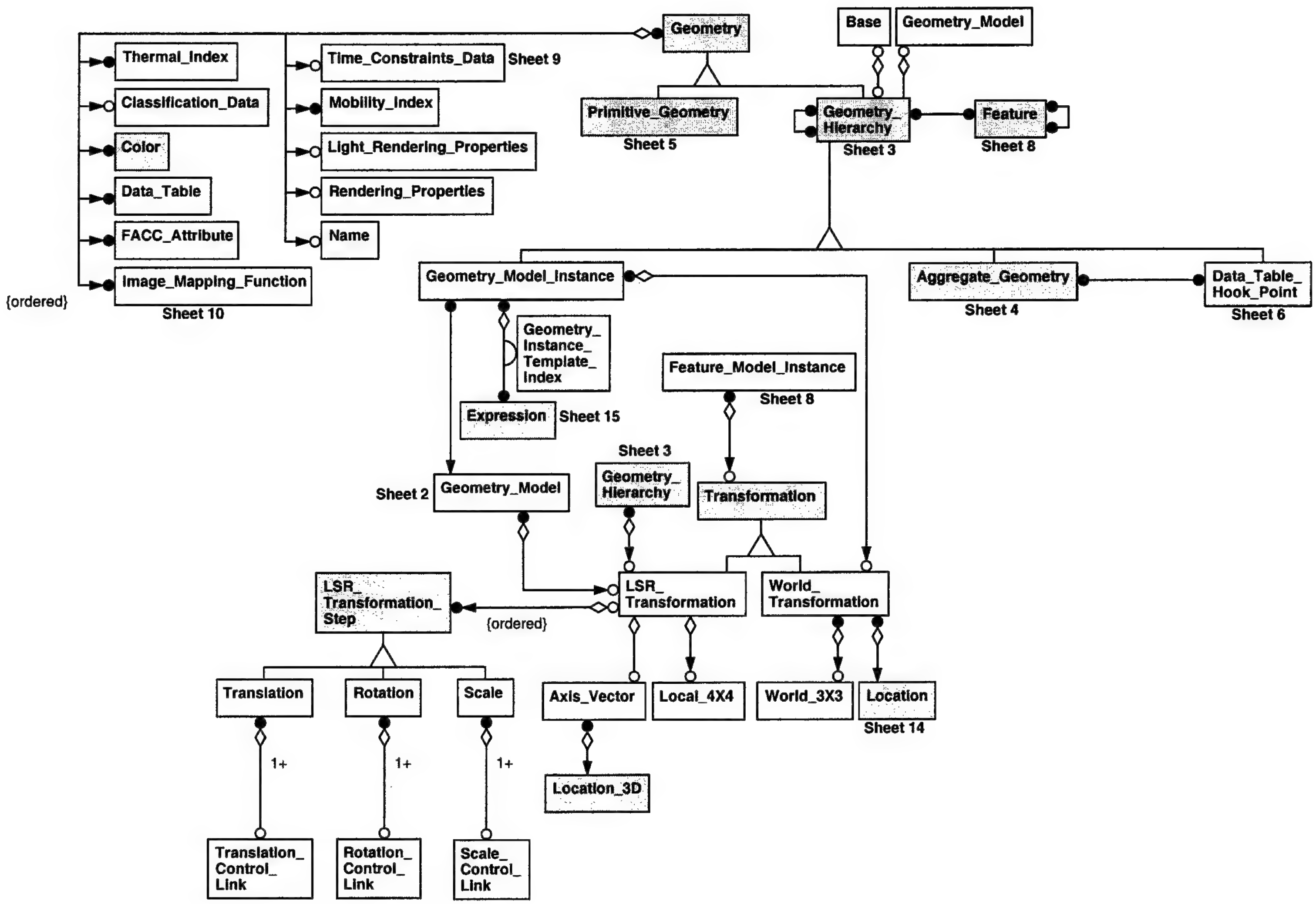
Sheet: 5: Primitive Geometry Panel: (0,0)



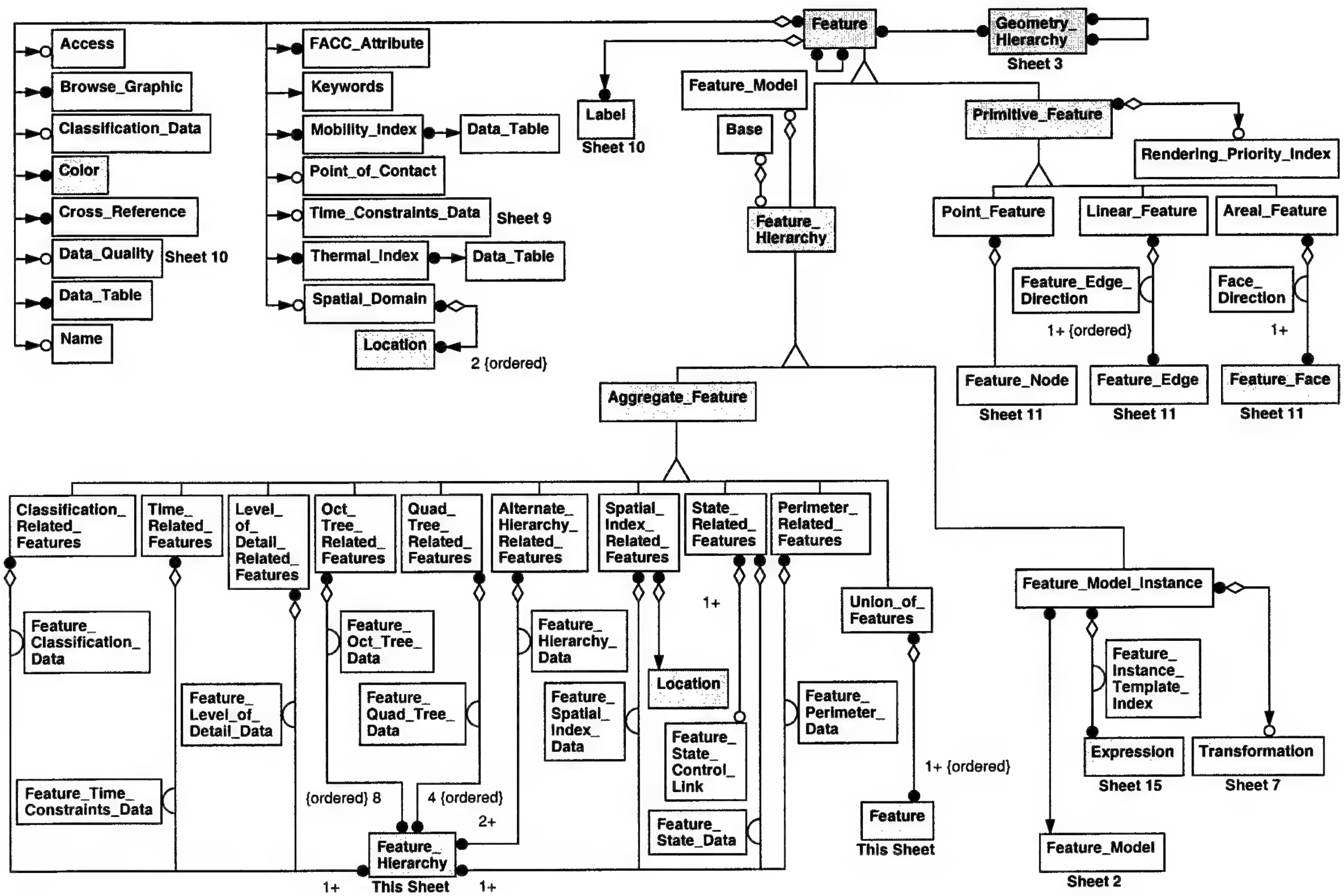
Sheet: 6: Data Table Panel: (0,0)



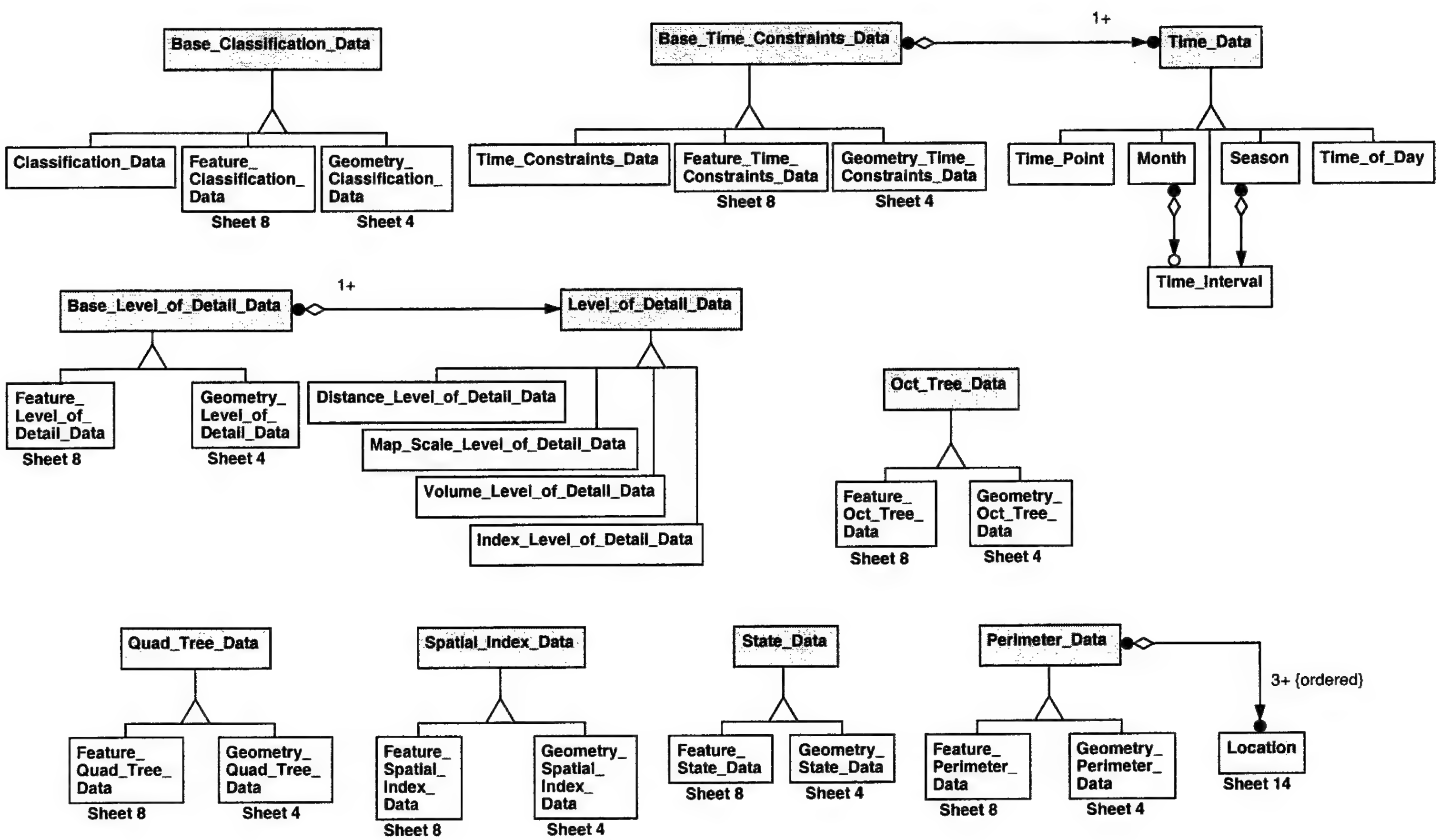
Sheet: 7: Geometry Connection and Transformation Panel: (0,0)



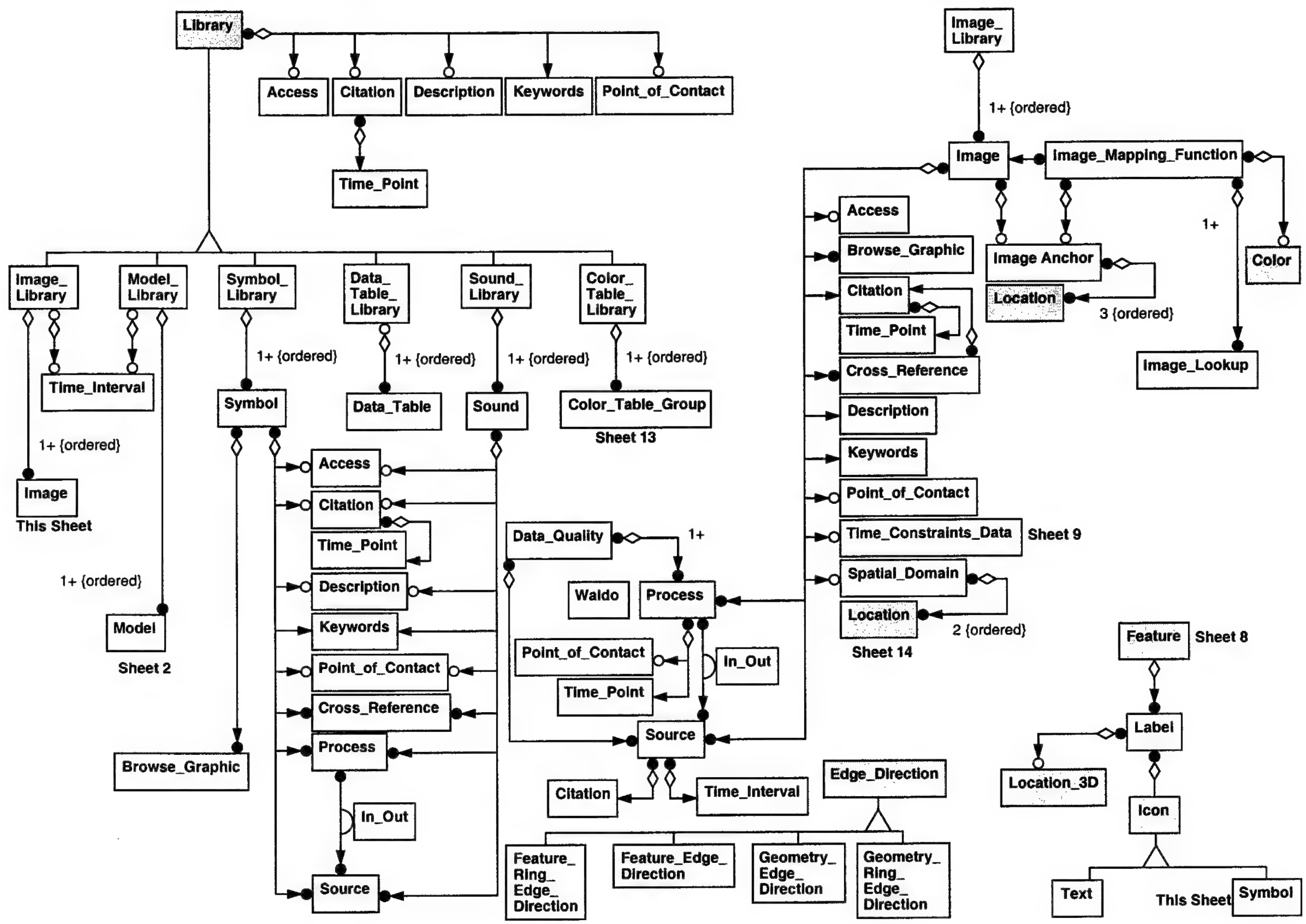
Sheet: 8: Feature & Aggregate Feature Panel: (0,0)



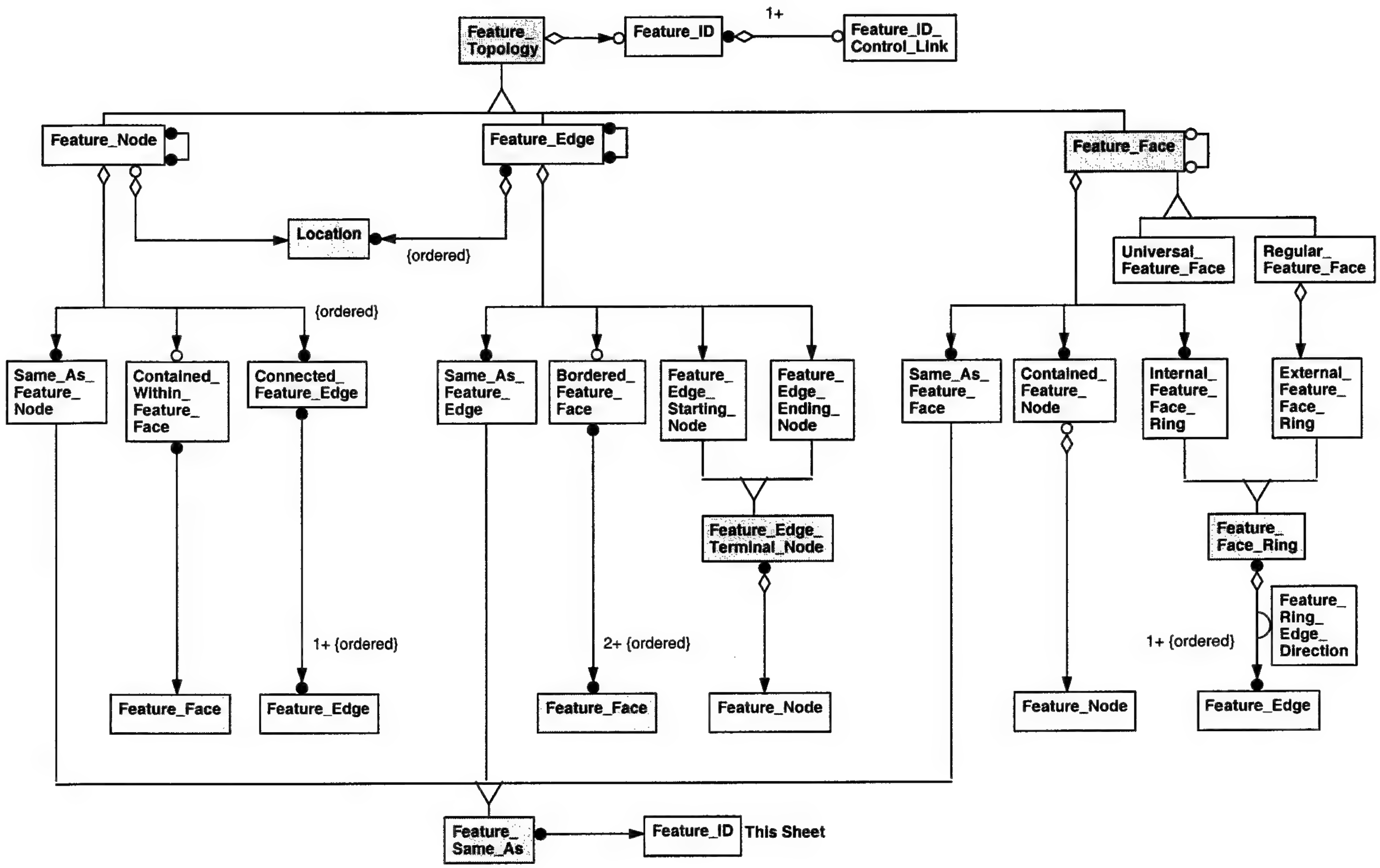
Sheet: 9: Base Data Classes Panel: (0,0)



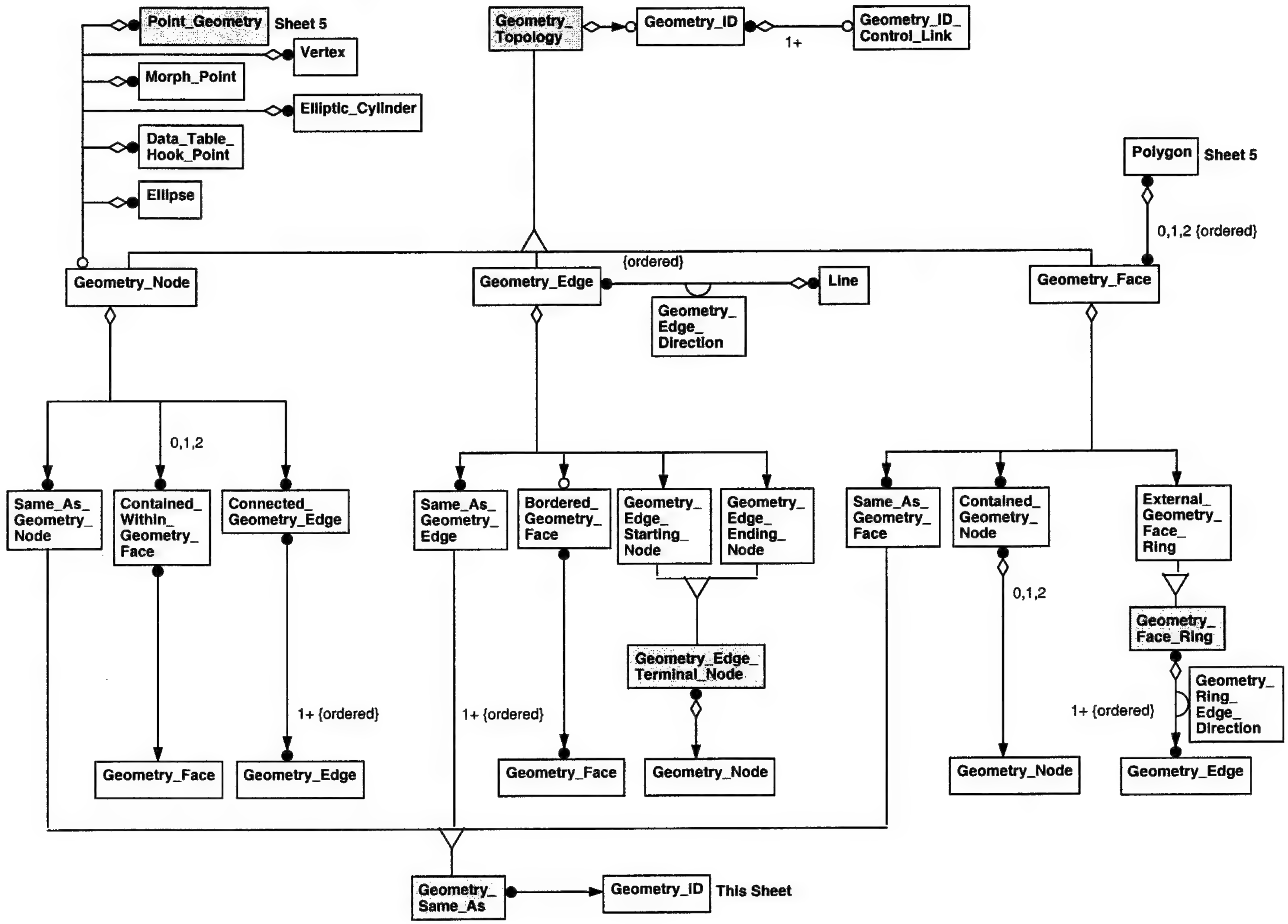
Sheet: 10: Libraries/Topology Tables/Edge Direction/Sound/Texture/Icon/Label Panel: (0,0)



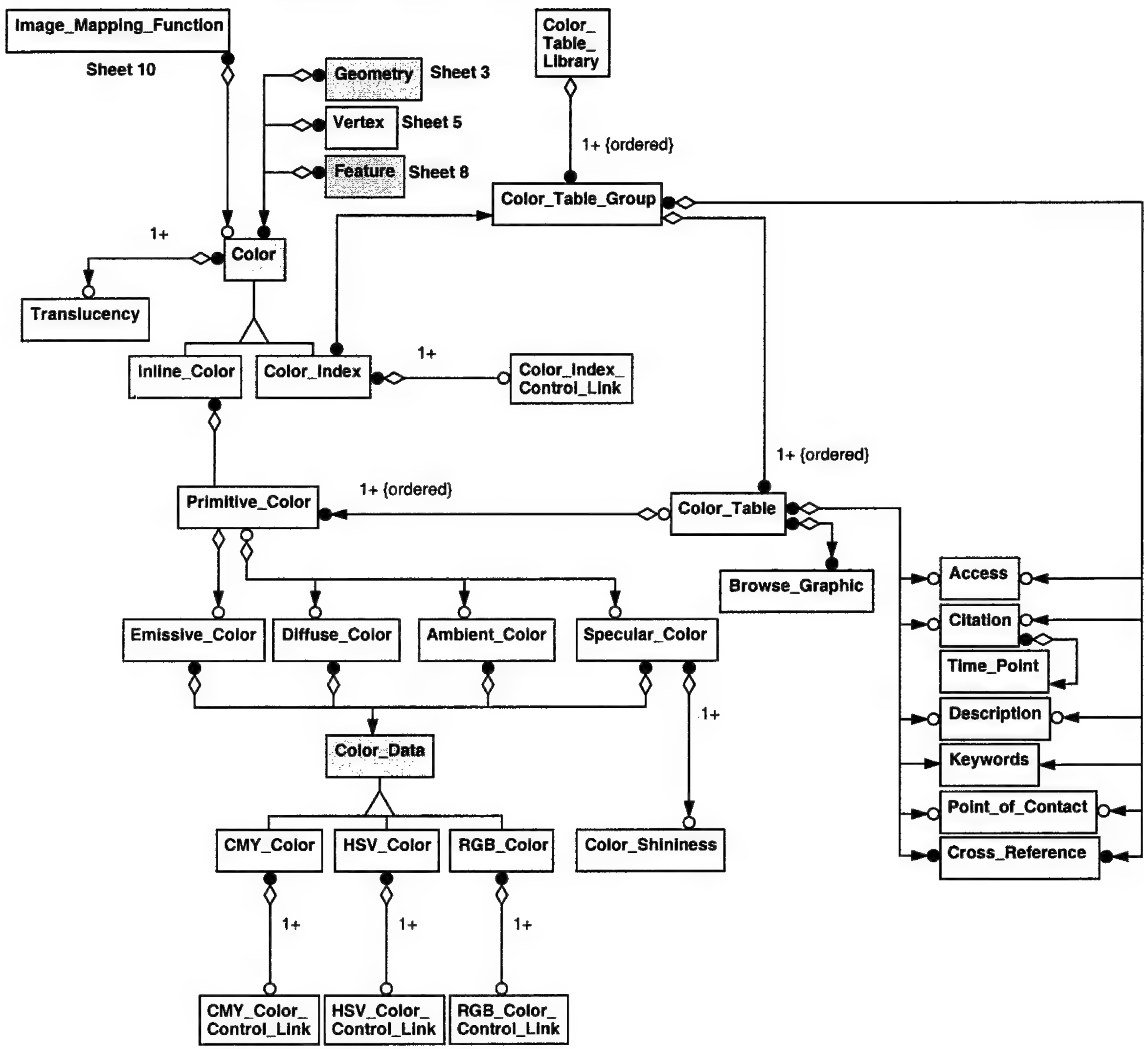
Sheet: 11: Feature Topology Panel: (0,0)



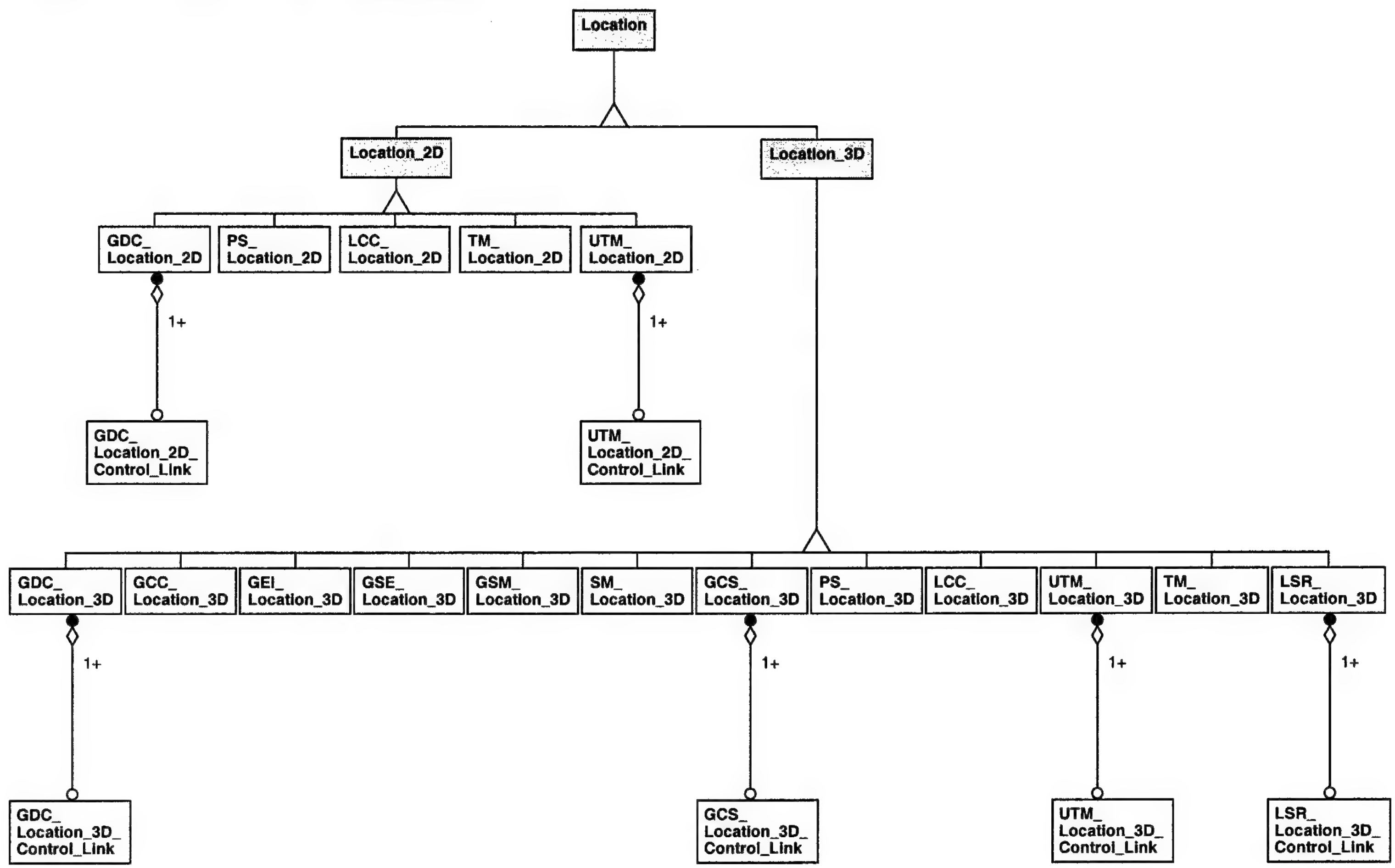
Sheet: 12: Geometry Topology Panel: (0,0)



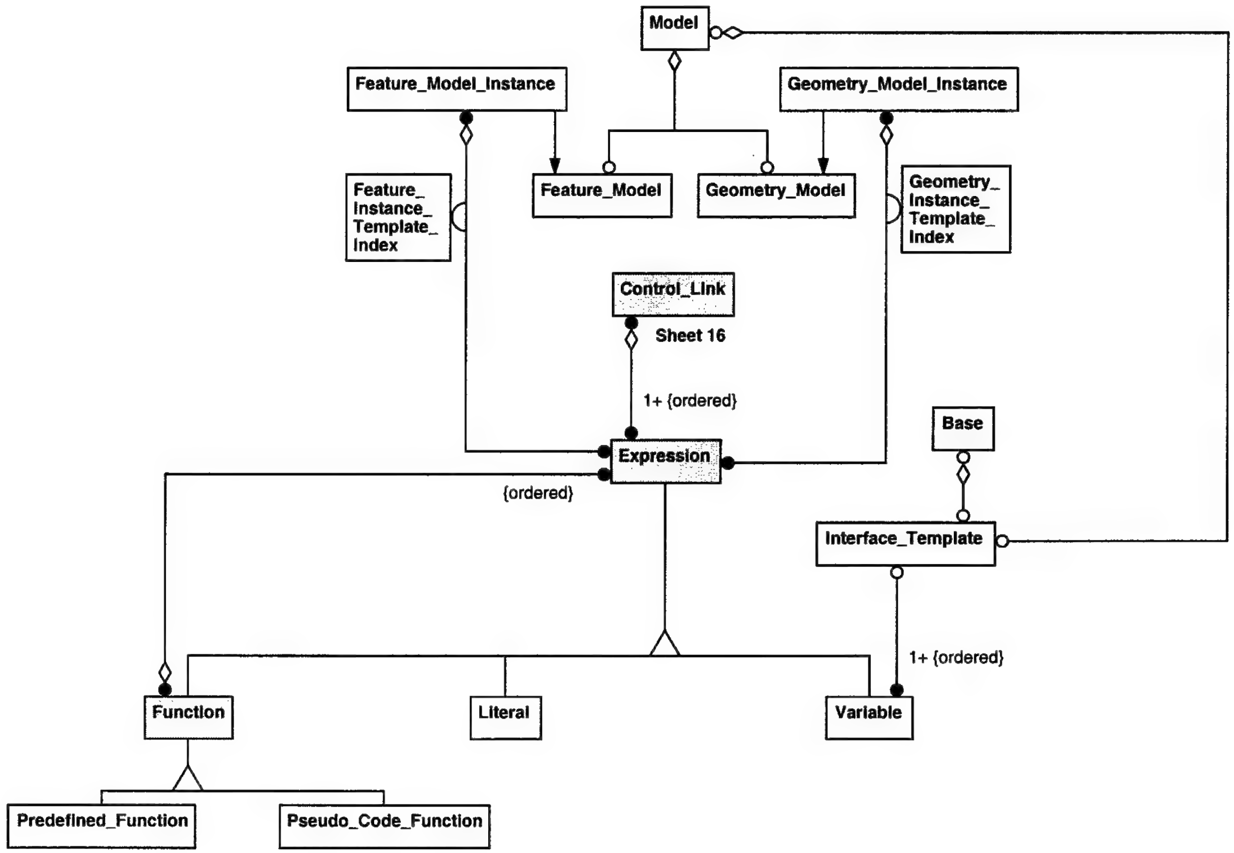
Sheet: 13: Color Tables Panel: (0,0)



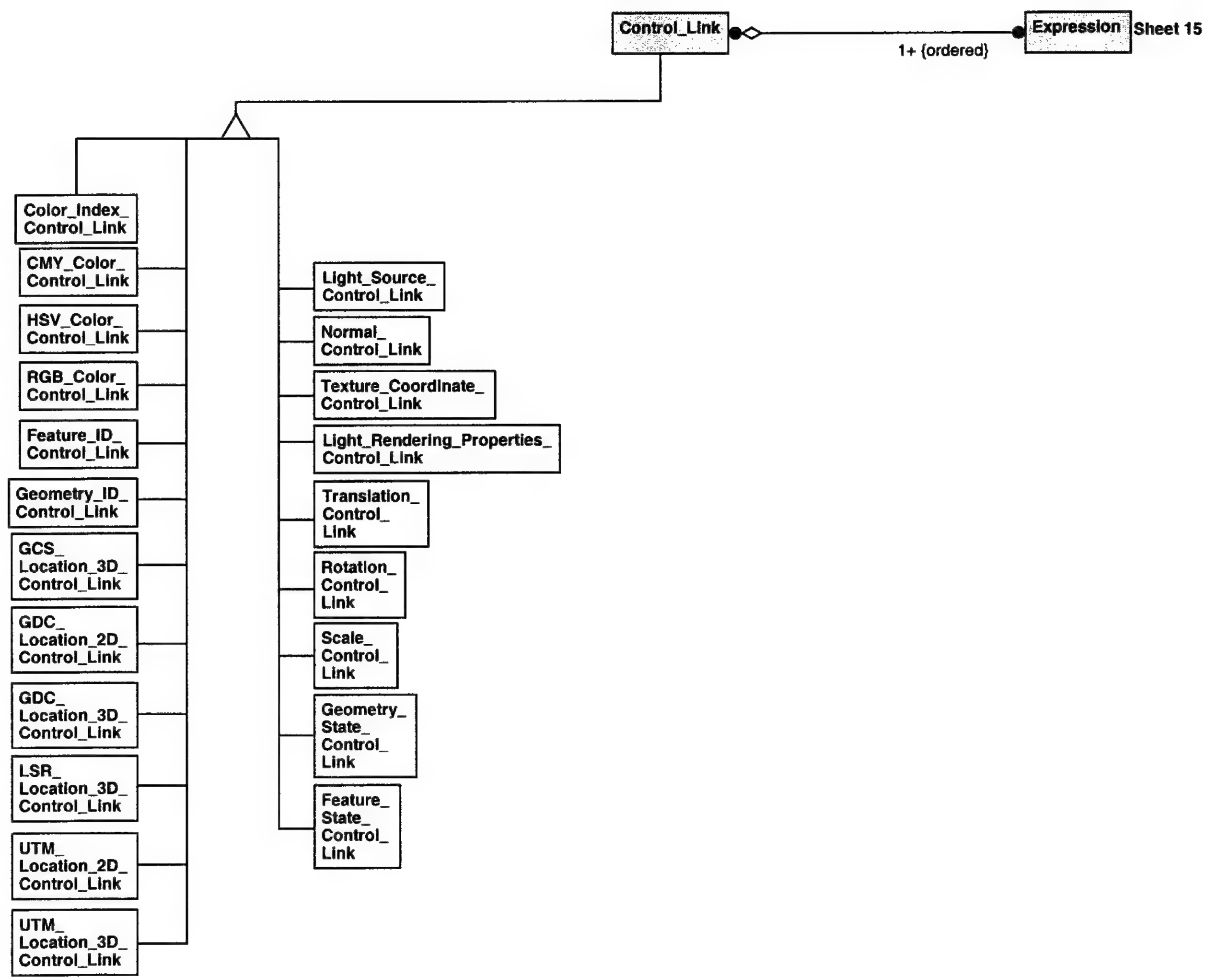
Sheet: 14: Location 3D Panel: (0,0)



Sheet: 15: Expressions Panel: (0,0)



Sheet: 16: Control Link Subclasses Panel: (0,0)



Appendix C

The SEDRIS Team

The SEDRIS Team

This appendix provides a detailed explanation of the composition of the SEDRIS Team and the responsibilities of the sponsoring agencies and associate contractors. The information in this appendix comes from Volume 2 of STRICOM's (1998b) documentation set titled *Synthetic Environment Data Representation ad Interchange Specification Overview* and the SEDRIS Homepage.

The SEDRIS Team consists of the DoD sponsoring agencies and a development team of contractors. The development effort is organized with contractors and government technical experts participating in a collaborative arrangement designed to focus the best available talent on data model and support software design.

DoD Sponsoring Agencies

SEDRIS addresses a broad spectrum of database interchange issues that cut across the military service areas of responsibility within the DoD. Additionally, the vast majority of knowledge in database interchange issues resides outside the DoD in industry. This situation necessitates a novel management approach that focuses government investment and priorities but takes advantage of industry partners' expertise. The management approach implemented in SEDRIS provides DoD oversight, while delegating implementation issues to specific associate contractors. These contractors possess both the insight into the problem domain and the ability to implement effective

solutions by using on-going research and development as well as tapping their broad spectrum of expertise.

The principal DoD sponsoring agencies are:

- Defense Modeling and Simulation Office (DMSO) - DMSO is the resource sponsor for the SEDRIS program. The DMSO Environmental Division focuses program direction with program management executed through the designated environmental DoD M&S Executive Agent Offices:
  - Terrain Modeling Project Office (TMPO), executive agent for terrain,
  - Ocean Executive Agency (OEA), executive agent for ocean,
  - Air and Space Natural Environment (ASNE), executive agent for air and space natural environment.
- Simulation Training and Instrumentation Command (STRICOM) - U.S. Army STRICOM provides the technical oversight and contract management support to the SEDRIS Program. STRICOM receives support from both internal assets (Engineering and Contracts), as well as external agencies (Naval Air Warfare Center - Training Systems Division).
- Defense Advance Research Projects Agency (DARPA) - DARPA provides support to the SEDRIS Program through the Information Systems Technology Office Program Manager for Synthetic Environments (PM-SE). PM-SE provides personnel resources for both the management and development of SEDRIS, as well as augmenting the contracting support provided by STRICOM.

Associate Contractors

The role of the SEDRIS associate contractors has been to channel their vast experience into the data model refinement process. Their major effort has involved performing an analysis of the SEDRIS data model. This analysis focused on the ability of the data model to adequately represent their vendor-unique data. The vendors' crystallization of this analysis was documented in respective mapping documents. After a thorough data model analysis, each contractor developed access software, using the SEDRIS API with their native database formats, as a primary deliverable. This access software allows each vendor to produce SEDRIS data as well as consume data written by others. The SEDRIS associate contractors are listed below. More information can be found in the Who's Involved section at the SEDRIS Homepage (www.sedris.org) where there is a link to each associate contractor.

- A&T - Analysis and Technology, Inc.
- AcuSoft, Inc.
- AFTS – Armed Forces Training Systems, Inc.
- ATLAS Elektronik GmbH
- Centric Software, Inc.
- Cybernet Systems Corp.
- Defence Evaluation and Research Agency
- Environmental Systems Research Institute, Inc.
- Evans and Sutherland Computer Corp.
- JRM Enterprises, Inc.

- Litton Industries - TASC, Inc.
- Lockheed Martin Information Systems
- Lockheed Martin Tactical Defense Systems
- The MITRE Corporation
- MultiGen - Paradigm, Inc.
- Naval Research Laboratory
- PAR Government Systems Corporation
- Raytheon Systems Company
- Reality by Design, Inc.
- SAIC - Science Applications International Corporation
- Silicon Graphics, Inc.
- SRI International
- TerraSim, Inc.
- Thomson Training & Simulation

Appendix D

Sensor Simulation/Tool to SEDRIS Mapping Document

SEDRIS Attribute and Description	Sampled Sensor Simulation/Tool	Sampled Sensor Simulation/Tool Term(s)
Reflectivity: Ratio of reflected (specularly, diffusely, or otherwise) flux divided by incident flux. (unitless)	RADSIM™	Modifiable reflectivity values Leading edge enhancement Far shore brightening
	SensorVision™	Reflection from sun
	Radar Toolkit™	Far shore brightening
	SOF ATS	Reflectivity
	TTS BDD3	Reflectivity
	IRGen®	Solar reflectivity
Diffuse_Reflectivity: Diffused component of reflectivity; diffusely reflected flux divided by incident flux; diffuse reflection pertains to the manner in which light is reflected and scattered as by a rough surface (or a Lambertian reflectors). (unitless)	Irma	Diffuse reflections Angle-dependent scattering
Specular_Reflectivity: Specular component of reflectivity; specularly reflected flux divided by incident flux; specular reflection pertains to the manner in which light is reflected, as by a mirror. (unitless)	Irma	Specular reflections Angle-dependent specular solar glints
BRDF (Bi-directional Reflectance Distribution Function): Ratio of reflected radiance to the incident irradiance; or reflectivity per solid angle; BRDF is a function of both angles of incidence and of reflection, and has units of reciprical Steradian. (1/Sr)	Radar Toolkit™	Target aspect
	TTS BDD3	Directionality and reflectivity
	Irma	Spectral response profiles

SEDRIS Attribute and Description	Sampled Sensor Simulation/Tool	Sampled Sensor Simulation/Tool Term(s)
Emissivity: Ratio of the emission of a sample to that of an ideal blackbody at the same temperature and in the same spectral interval. (unitless)	SensorVision™	Thermal emission
	CCTT	Emissivity
	Irma	Thermal emissions Angle-dependent surface emission
	TTS BDD3	Emissivity (directivity)
	STM	Infrared radiance
	IRGen®	Long-IR emissivity
Transmissivity: Ratio of transmitted flux to incident flux per kilometer; note that this includes direct as well as scattered transmission and therefore is not necessarily related to transmission loss; i.e., transmissivity plus transmission_loss does not necessarily equal unity. (1/km)	CCTT	Atmospheric transmittance
	RADSIM™	Cultural feature shadowing Terrain masking and terrain following
	Irma	Path transmittance effects Shadowing
	SOF ATS	Transparency
	TTS BDD3	3D feature shadowing Alpha (transparency)
	IRGen®	Cloud transmission
Total_Transmissivity: Ratio of transmitted flux to incident flux through an object. (unitless)	Radar Toolkit™	Radar shadowing Terrain/feature/target masking
	TTS BDD3	3D feature shadowing Alpha (transparency)
Transmission_Loss: Ratio of flux lost (due to absorption, scattering, or otherwise) to total flux from one traversal through the medium or object. (dB)	Radar Toolkit™	Range attenuation
	TTS BDD3	Reflectivity attenuation

SEDRIS Attribute and Description	Sampled Sensor Simulation/Tool	Sampled Sensor Simulation/Tool Term(s)
Transmission_Attenuation: Ratio of flux lost (due to absorption, scattering, or otherwise) per km to total flux from one traversal through the medium or object. (dB/km)	Radar Toolkit™	Atmospheric attenuation
	RADSIM™	Range attenuation
	CCTT	Mass extinction coefficients for typical obscurants, WP, & RP smokes (α) Ambient attenuation (k)
	SensorVision™	Atmospheric attenuation
Surface_Backscatter: Ratio of flux reflected directly back to the direction of incidence, to incident flux. (unitless)	RADSIM™	Background texture and noise
	Radar Toolkit™	Background noise
	Irma	Temporal correlation in clutter
	STM	Radar backscatter
Volume_Backscatter: Ratio of flux, per km of range, reflected directly back to the direction of incidence, to incident flux. (1/km)	SensorVision™	Path radiance
	Radar Toolkit™	Atmospheric noise
	RADSIM™	Chaff Three-dimensional thunderstorm effects
	Irma	Path radiance effects
Radar_Significant_Factor (RSF): Fourteen, or more, homogenous groupings categorizing manmade or natural object based on the object's predominant exposed surface material (see DFAD spec). (enum)	Radar Toolkit™	Feature ID Code (FIC) (RSF)
Ground_Clutter: Ground clutter expressed in average radar cross section per unit area of land. (unitless)	Radar Toolkit™	Ground clutter
	Irma	Clutter
Sea_Clutter: Sea clutter expressed in average radar cross section per unit area of sea. (unitless)	Radar Toolkit™	Sea clutter Sea state
	IRMA	Clutter

SEDRIS Attribute and Description	Sampled Sensor Simulation/Tool	Sampled Sensor Simulation/Tool Term(s)
Radar_Cross_Section (RCS): Power (or flux) reflected from the target toward the source per solid angle divided by incident power density times 4 Pi; or fictional area intercepting that amount of power, which, when scattered equally in all direction, produces an echo at the radar equal to that from the target. (sq meters)	Radar Toolkit™	Radar cross section (RCS)
	STM	Radar cross section with Polarization (HH,VV,VH, VV) and incident angle
Absorptivity: Ratio of absorbed flux to the incident flux. (unitless)	CCTT	Absorptance
Solar_Absorptivity: Ratio of absorbed flux to the incident solar flux. (unitless)	Irma	Direct and diffuse solar heating
Thermal_Contrast: Contrast of the target at the target location expressed as difference in temperature of target to background. Note: this term is dependent on location, time, and background. (degress Kelvin)	SensorVision™	Background feature temperature
	CCTT	Thermal contrast
	TTS BDD3	Delta_T
	Irma	Interpolated temperatures
Visual_Contrast: Contrast of the target at the target location expressed as a number between 0 and 1. Also dependent on location, time, and background. (unitless)	CCTT	Visual contrast
Fine_Scale_Surface_Roughness: The fine scale roughness (i.e., about the size of the EM radiation wavelength of concern), expressed as the standard deviation of the surface variation. (meters)	Irma	Fine scale surface roughness
	IR Posse EO/IR data dictionary	Fine scale roughness standard deviation

SEDRIS Attribute and Description	Sampled Sensor Simulation/Tool	Sampled Sensor Simulation/Tool Term(s)
Fine_Scale_Roughness_Correlation_ Length: Correlation length of the fine scale roughness. (meters)	Irma	Fine scale roughness correlation length
	IR Posse EO/IR data dictionary	Fine scale roughness correlation length
Large_Scale_Surface_Roughness: The large scale roughness (i.e., orders of magnitude larger than the EM radiation wavelength of concern), expressed as the standard deviation of the surface variation. (meters)	Irma	Large scale surface roughness
	IR Posse EO/IR data dictionary	Large scale roughness standard deviation
Large_Scale_Roughness_Coefficient_ Length: Correlation length of the large scale roughness. (meters)	Irma	Large scale roughness correlation length
	IR Posse EO/IR data dictionary	Large scale roughness correlation length
Secondary_Texture: Value with range from 0.0 to 1.0 indicating the prevalence of an image's visual texture is in the respective sensor domain, with 1.0 representing most predominant. (unitless)	CCTT	Secondary color (texture) scalar
Sun_Shading: Scalar value from 0.0 to 1.0 indicating significance of the effects of sun shading with 1.0 representing most effective. (unitless)	CCTT	Sun shading scalar
Radiance: Radiance of an object. (Watts per sq meter per steradian)	SOF ATS	Luminosity
	STM	Infrared radiance
	SensorVision™	Radiance
	Irma	Solar radiance
Radiance_Amplitude: Value from 0 to 255 indicating the amount of fluctuation of the material's radiance (or luminance) over a 24-hour period. (integer)	TTS BDD3	Luminance amplitude

SEDRIS Attribute and Description	Sampled Sensor Simulation/Tool	Sampled Sensor Simulation/Tool Term(s)
Radiance_Phase: Value from 0 and 24 indicating phase of the Radiance_ Amplitude, assuming a sine wave fluctuation. (units/hr)	TTS BDD3	Phase of luminance amplitude
Relative_Radiance: Value from 0 to 255 indicating relative radiance (or luminance) with 0 representing coldest and 255 representing hottest. (integer)	TTS BDD3	Relative luminance
Real_Refraction_Index: Real part of refraction index. (unitless)	Radar Toolkit™	Earth curvature effects
	RADSIM™	Earth curvature
Imaginary_Refraction_Index: Imaginary part of refraction index. (unitless)	TMPO (Dr. Cornette)	Imaginary refraction index
Irradiance: The incident energy on a surface or onto specific objects. (Watts per sq meter)	TMPO (Dr. Cornette)	Irradiance
Direct_Solar_Radiance: Direct solar irradiance. (Watts per sq meter)	Irma	Sun illumination
	IR Posse EO/IR data dictionary	Total direct solar insolence
Diffused_Solar_Radiance: Diffused solar irradiance. (Watts per sq meter)	SensorVision™	Reflection from sun
	IR Posse EO/IR data dictionary	Total diffused solar insolence
Direct_Lunar_Radiance: Direct lunar irradiance. (Watts per sq meter)	IR Posse EO/IR data dictionary	Total direct lunar luminance
Diffused_Lunar_Radiance: Diffuse lunar irradiance. (Watts per sq meter)	IR Posse EO/IR data dictionary	Total diffused lunar luminance

SEDRIS Attribute and Description	Sampled Sensor Simulation/Tool	Sampled Sensor Simulation/Tool Term(s)
Direct_Downwelling: Irradiance from the sky onto the particular object. (Watts per sq meter)	Irma	Angle-dependent skyshine heating
	SensorVision™	Ambient sky illumination
Light_Level: Light level, or illuminance, or incidance (analogous to irrandiance) weighted by the response of the eye. (lux)	CCTT	Light levels
Thermal_Conductivity: Ratio of thermal flux density through a surface within a material to the temperature gradient across the surface. (Watts per meter-K)	CCTT	Conductivity
	Irma	Thermal conduction
	IRGen®	Thermal conductivity
	IR Posse EO/IR data dictionary	Thermal conductivity
Density: Mass per unit volume. (kg per cubic meter)	IRGen®	Density
	IR Posse EO/IR data dictionary	Density
Specific_Heat: Quantity of heat required to raise the temperature of a unit mass of the material one degree Kelvin. (joules/gram-K)	IRGen®	Specific heat
	IR Posse EO/IR data dictionary	Specific heat
Convection_Coefficient: Ratio of convective heat flux density through a surface boundary to the temperature gradient across that boundary. (unitless)	Irma	Free and forced convection
	IR Posse EO/IR data dictionary	Coefficient of convection
Diurnal_Depth: Thermal thickness; effective depth of temperature variation over diurnal cycle. (m)	CCTT	Diurnal depth
	IR Posse EO/IR data dictionary	Diurnal depth

SEDRIS Attribute and Description	Sampled Sensor Simulation/Tool	Sampled Sensor Simulation/Tool Term(s)
Thermal_Mass: Responsiveness (resistance) of an object to heat; or density times specific heat capacity times thermal thickness. (W/sq m/deg K)	IR Posse EO/IR data dictionary	Thermal mass
Thickness: Physical thickness. (m)	IRGen®	Thickness
	TMPO (Dr. Cornette)	Material thickness
Interior_Temperature: Temperature of interior of the surface material. (K)	IRMA	Interior heating
	IRGen®	Interior temperature
	IR Posse EO/IR data dictionary	Interior/Support temperature
Internal_Flow_Velocity: Speed of any internal convective flow. (m/sec)	IRGen®	Interior flow velocity
	IR Posse EO/IR data dictionary	Interior flow velocity
Support_Temperature_Code: An enumeration indicating the material's (or object's) temperature. Same codes used. (enum)	CCTT	Support temperature codes 1) significantly influenced by ambient air, 2) significantly influenced by ground temperature, 3) artificially heated or cooled to a steady state temperature, 4) artificially heated or cooled to room temperature 5) warmed by trapping heat or being near hot heat sources, 6) influenced by heat generating engines, and 7) is forced to extremely hot temperature
Maximum_Temperature: Maximum temperature a material or object can achieve. (K)	CCTT	Maximum temperature

Appendix E

SEDRIS Change Request Number: A&T-017

SEDRIS Change Request

SCR #: A&T-017 Units

Date submitted: 15 June 1998

Action Required date: Next Data Model Release

Problem: 1. SCR #: SE_CORE-117 defines Property attribute "unit" in terms of undefined SE_UNIT_ENUM.
2. Values in class Axis are unit ambigous.

Solution (proposal):
1. Define SE_UNIT_ENUM as:

```
/*
 * SE_UNIT_ENUM
 *
 * Unit designations for <Axis> axis_unit and <Property> value_unit
 *
 * Guidelines for extensions to SE_UNIT_ENUM:
 *  1) Use SI units if at all possible.
 *  2) Use scaled SI units if customary in the community.
 *       Example:
 *           Pascal - SI unit
 *           hectoPascal - customary in Atmospheric community
 *           microPascal - customary in Acoustic cummunity
 *  3) Use non-SI unit ONLY if WIDELY accepted in the user community,
 *      otherwise convert your units to somthing resonable for the consumer.
 *  4) Use "native" units only for MODEL specific tables;
 *       Example:
 *           SE_TABLE_COMBIC_CLOUD_HISTORY, and
 *           SE_TABLE_ACOUSTIC_LFBL_PARAMETERS
 *           are two model specific data tables. Users expect the
 *           units to match the algorithm inputs.
 * Some units have been included for compatability with DIGEST FACC 2.0
 * (e.g. SE_UNITS_HECTARES), but their use should be strongly discouraged.
 *
 */
typedef enum
{
  SE_UNITS_NO_UNITS,          /* units not applicable */
  SE_UNITS_ENUMERATION,       /* for Enumeration Axis */
  SE_UNITS_SE_STRING,         /* string valued data */
```

```
SE_UNITS_COMPONENT_INDEX,  /* aggregated table index */
SE_UNITS_INDEX,            /* positive integer */

/*Ratio or pure number */
SE_UNITS_UNITLESS,
SE_UNITS_PERCENT,
SE_UNITS_PER_THOUSAND,
SE_UNITS_PARTS_PER_THOUSAND,
SE_UNITS_PARTS_PER_MILLION,
SE_UNITS_DECIBELS,

/*Time*/
SE_UNITS_MICRO_SECONDS,
SE_UNITS_SECOND,           /* SI unit */
SE_UNITS_MINUTE,
SE_UNITS_HOUR,
SE_UNITS_DAY,
SE_UNITS_JULIAN_DAY_NUMBER, /* 1=Jan 01 */
SE_UNITS_SE_SEASON_ENUM,
SE_UNITS_YEAR,

/*Frequency, Reciprocal time*/
SE_UNITS_HERTZ,            /* derived SI (1/s)*/
SE_UNITS_KILOHERTZ,
SE_UNITS_MEGAHERTZ,
SE_UNITS_GIGAHERTZ,

SE_UNITS_RECIPROCAL_SECONDS,

/*Angular measure*/
SE_UNITS_RADIAN,           /* derived SI */
SE_UNITS_ARC_DEGREE,
SE_UNITS_ARC_DEGREE_RELATIVE_TRUE_NORTH,
SE_UNITS_STERADIAN,        /* derived SI */

SE_UNITS_RECIPROCAL_STERADIAN,

/*Length*/
SE_UNITS_MICROMETER,
SE_UNITS_MILLEMETER,
SE_UNITS_CENTIMETER,
SE_UNITS_DECIMETER,
SE_UNITS_METER,            /* SI unit */
```

```
SE_UNITS_KILOMETER,
SE_UNITS_FEET,      /*FACC unit*/
SE_UNITS_NAUTICAL_MILE,    /*FACC unit*/

SE_UNITS_RECIPROCAL_METER,

/*Area*/
SE_UNITS_SQUARE_METER,
SE_UNITS_HECTARES,  /*FACC unit*/

/*Speed*/
SE_UNITS_METERS_PER_SECOND,  /* derived SI unit */
SE_UNITS_MILLIMETERS_PER_HOUR,
SE_UNITS_CENTIMETERS_PER_HOUR,
SE_UNITS_METERS_PER_HOUR,
SE_UNITS_KILOMETERS_PER_HOUR,
SE_UNITS_KNOTS,     /*FACC unit*/

/*Mass*/
SE_UNITS_GRAM,
SE_UNITS_KILOGRAM,         /* SI unit */
SE_UNITS_KIP,       /*FACC unit - KiloPound*/
SE_UNITS_TON,       /*FACC unit*/

/*force, work, power*/
SE_UNITS_NEWTON,           /* derived SI (Meter Kilogram/Second^2)*/
SE_UNITS_JOULE,          /* derived SI (Newton Meter)*/
SE_UNITS_WATT,           /* derived SI (Joule/Second)*/
SE_UNITS_MEGAWATT,  /*FACC unit*/

/*Temperature*/
SE_UNITS_DEGREES_CELSIUS,
SE_UNITS_DEGREES_KELVIN,    /* SI unit*/

/*Pressure*/
SE_UNITS_MICROPASCAL,
SE_UNITS_PASCAL,          /* derived SI (Newton/Meter^2)*/
SE_UNITS_HECTOPASCAL,

/*Density*/
SE_UNITS_KILOGRAM_PER_CUBIC_METER,
SE_UNITS_GRAM_PER_CUBIC_CENTIMETER,

/*Electric units*/
```

```
SE_UNITS_AMPERE,          /* SI unit */
SE_UNITS_KILOAMPERE,
SE_UNITS_VOLT,            /*derived SI (Watt/Amp)*/
SE_UNITS_KILOVOLT,
SE_UNITS_OHM,             /*derived SI (Volt/Amp)*/
SE_UNITS_SIEMANS,         /*derived SI (1/Ohm)*/
SE_UNITS_WEBER,           /*derived SI (Volt Second)*/
SE_UNITS_NANOTESLA,
SE_UNITS_TESLA,           /*derived SI (Weber/Meter^2)*/

/*Illumination */
SE_UNITS_CANDELA,         /* SI unit */
SE_UNITS_LUMEN,           /* derived SI (Candela Steradian)*/
SE_UNITS_MICROLUX,
SE_UNITS_LUX,             /* derived SI (Lumen/Meter^2)*/

/*Miscellanous in alphabetic order*/
SE_UNITS_BOOLEAN,
SE_UNITS_DEGREES_CELSIUS_PER_HOUR, /*radiation phenology?*/
SE_UNITS_DEGREES_CELSIUS_PER_METER, /*temperature gradiant*/
SE_UNITS_DEGREES_CELSIUS_PER_KILOMETER,
SE_UNITS_DECIBELS_REFERENCE_ONE_MICROPASCAL_AT_ONE_METER,
SE_UNITS_DECIBELS_REFERENCE_ONE_MICROPASCAL_PER_HERTZ,

SE_UNITS_DECIBELS_REFERENCE_ONE_MICROPASCAL_PER_HERTZ_PER_A
RC_DEGREE,
SE_UNITS_DECIBELS_REFERENCE_ONE_SQUARE_METER,
SE_UNITS_DECIBELS_PER_OCTAVE,
SE_UNITS_DECIBELS_PER_METER_PER_HERTZ,
SE_UNITS_DECIBELS_PER_METER_KILOHERTZ,
SE_UNITS_DECIBELS_PER_SQUARE_METER,
SE_UNITS_DECIBELS_PER_SQUARE_METER_PER_HERTZ,
SE_UNITS_HECTOPASCAL_PER_SECOND,
SE_UNITS_JOULE_PER_GRAM_KELVIN,    /*specific heat*/
SE_UNITS_KILOGRAM_PER_SQUARE_METER,
SE_UNITS_PH_LEVEL,
SE_UNITS_PHOTONS_PER_CUBIC_CENTIMETER_PER_SECOND,
SE_UNITS_PIXEL,
SE_UNITS_SIEMANS_PER_METER,
SE_UNITS_SQUARE_METERS_PER_HERTZ_RADIAN,/*wave spectrum?*/
SE_UNITS_TEXEL,
SE_UNITS_WATT_PER_METER,           /*dissipation rate*/
SE_UNITS_WATT_PER_SQUARE_METER,    /*heat flux*/
SE_UNITS_WATT_PER_METER_KELVIN,    /*thermal conductivity*/
```

SE_UNITS_WATT_PER_SQUARE_METER_KELVIN,/*thermal mass*/

/* Algorithm Specific */
SE_UNITS_ALGORITHM_SPECIFIC

} SE_UNIT_ENUM;

Policy guide for SE_UNIT_ENUM maintainence:
1) All SI units are allowable;
2) Scaled (powers of 10) SI _if_ in wide spread use in consumer community.
3) FACC units (to be not FACC inconsistent)
4) Model/Algorithm specific units can use SE_UNITS_ALGORITHM_SPECIFIC if diss-allowed by 1,2,3

2. Add SE_UNIT_ENUM axis_unit attribute to Axis class.

Disposition (approve, disapprove, ...)
Rationale:
Use of unit attributes for Axis and Property classes agreed upon by OATS/SAM #8.

Disposition Date:

Disposition Authority:

SEDRIS CHIEF SOFTWARE ENGINEER Date

Appendix F

SEDRIS Change Request Number: A&T-019

SEDRIS Change Request

SCR #: A&T-019 revision 2

Date submitted: 29 June 1998
Date revised: 01 July 1998
Revised to show:
New SE_DATA_TABLE_VALUE_TYPE_ENUM entries,
New struct SE_DATA_TABLE_VALUE.u entries,
New struct SE_DATA_TABLE_VALUE_PTR.u entries,
and added comments (and fixed spelling: ABSORPTIVITY)

Action Required date: Next Data Model Release

Problem: New Data Table types and data value enumerations are needed to support Electro-magnetic and other physical properties of materials.

Solution (proposal):
Add the following -

New Enum Types:

```
/*
 * ENUM: SE_DT_POLARIZATION_ENUM
 *
 * Denote type of polarization in EM or light radiation.
 * Polarization is generally an axis in an EM (electromagnetic)
 * table type.  As an example, a SE_TABLE_EM_REFLECTIVITY table type
 * may have SE_DT_BRDF as one of its labels and SE_DT_POLARIZATION_ENUM
 * as one of its axes. This table would indicate that BRDF is a
 * function of EM polarization, i.e., reflectance can take on different
 * values for random, circular, crossed, or any other enumerated type of
 * polarization.
 */

typedef enum {
 SE_POLARIZATION_ALL, /* All or any
 SE_POLARIZATION_RANDOM, /* Random */
 SE_POLARIZATION_CIRCULAR, /* Circular */
 SE_POLARIZATION_ELLIPTICAL, /* Elliptical */
 SE_POLARIZATION_HH, /* Horizontal */
 SE_POLARIZATION_VV, /* Vertical) */
```

```
 SE_POLARIZATION_HV, /* Crossed */
 SE_POLARIZATION_VH, /* Crossed */
 SE_POLARIZATION_S,  /* S (perpendicular to incidence-reflectance plane) */
 SE_POLARIZATION_P /* P (parallel to incidence-reflectance plane)*/
} SE_DT_POLARIZATION_ENUM;


/*
 * ENUM: SE_DT_RADAR_SIGNIFCANT_FACTOR_ENUM
 *
 * Radar signfificant factor (RSF) is generally a label in a
 * SE_TABLE_EM_BACKSCATTER table type.
 */

typedef enum {
 SE_DT_RSF_METAL,
 SE_DT_RSF_PART_METAL,
 SE_DT_RSF_STONE,
 SE_DT_RSF_COMPOSITION,
 SE_DT_RSF_EARTHEN,
 SE_DT_RSF_WATER,
 SE_DT_RSF_SAND,
 SE_DT_RSF_ROCK,
 SE_DT_RSF_CONCRETE,
 SE_DT_RSF_OIL,
 SE_DT_RSF_MARSH,
 SE_DT_RSF_TREES,
 SE_DT_RSF_ICE,
 SE_DT_RSF_ASPHALT
} SE_DT_RADAR_SIGNIFCANT_FACTOR_ENUM;


/*
 * ENUM: SE_DT_EM_BAND_ENUM
 *
 * Denote a standard frequency band for EM emissions.
 * Electromagnetic band is generally an axis in an EM (electromagnetic)
 * table type.   As an example, a SE_TABLE_EM_REFLECTIVITY table type
 * may have SE_DT_BRDF as one of its labels and SE_DT_EM_BAND_ENUM
 * as one of its axes.  This table would indicate that BRDF is a
 * function of EM wavelength band, i.e., reflectance can take on
 * different values for IR, RADAR, or any other enumerated type of
 * wavelength band.   */
```

```
typedef enum {
 SE_EM_BAND_RF,        /* Radio frequency */
 SE_EM_BAND_ELF,       /* Extremely Low frequency */
 SE_EM_BAND_VLF,       /* Very Low frequency */
 SE_EM_BAND_LF,        /* Low frequency */
 SE_EM_BAND_MF,        /* Medium frequency */
 SE_EM_BAND_HF,        /* High frequency */
 SE_EM_BAND_VHF,       /* Very High frequency */
 SE_EM_BAND_UHF,       /* Ultra High frequency */
 SE_EM_BAND_SHF,       /* Super High frequency */
 SE_EM_BAND_EHF,       /* Ultra High frequency */
 SE_EM_BAND_MICROWAVE,
 SE_EM_BAND_P_BAND,    /* P-W  Radar bands */
 SE_EM_BAND_L_BAND,
 SE_EM_BAND_S_BAND,
 SE_EM_BAND_X_BAND,
 SE_EM_BAND_K_BAND,
 SE_EM_BAND_Q_BAND,
 SE_EM_BAND_V_BAND,
 SE_EM_BAND_W_BAND,
 SE_EM_BAND_IR,        /* Infrared bands */
 SE_EM_BAND_FAR_IR,
 SE_EM_BAND_NEAR_IR,
 SE_EM_BAND_UV,        /* Ultraviolet bands */
 SE_EM_BAND_NEAR_UV,
 SE_EM_BAND_FAR_UV,
 SE_EM_BAND_XRAY
} SE_DT_EM_BAND_ENUM;
/* note: EM wavelength band enumeration is already requested in SCR#
  PUBLIC-102.  This SCR also provides suggested names  */
```

New Data Table Types - SE_DATA_TABLE_TYPE_ENUM to be converted to SCC:

```
SE_TABLE_EM_ABSORPTIVITY,
SE_TABLE_EM_BACKSCATTER,
SE_TABLE_EM_CONTRAST,
SE_TABLE_EM_EMISSIVITY,
SE_TABLE_EM_IRRADIANCE,
SE_TABLE_EM_RADAR_CROSS_SECTION,
SE_TABLE_EM_RADIANCE,
SE_TABLE_EM_REFLECTIVITY,
SE_TABLE_EM_REFRACTION,
SE_TABLE_EM_SUN_SHADING,
SE_TABLE_EM_SURFACE_ROUGHNESS,
```

SE_TABLE_EM_SECONDARY_TEXTURE,
SE_TABLE_EM_TRANSMISSIVITY,
SE_TABLE_THERMOPHYSICAL

New Data Table Labels - SE_DATA_TABLE_LABLE_ENUM to be converted to SAC:

SE_DT_ABSORPTIVITY,
SE_DT_BRDF,
SE_DT_CLUTTER_GROUND,
SE_DT_CLUTTER_SEA,
SE_DT_CONVECTION_COEFFICIENT,
SE_DT_DIFFUSE_REFLECTIVITY,
SE_DT_DIRECT_DOWNWELLING,
SE_DT_DIURNAL DEPTH,
SE_DT_EM_BAND_LABEL,
SE_DT_FINE_SCALE_CORRELATION,
SE_DT_FINE_SCALE_ROUGHNESS,
SE_DT_IM_REFRACTIVE_INDEX,
SE_DT_INCIDENT_AZIMUTH,
SE_DT_INCIDENT_ELEVATION,
SE_DT_INERIOR_TEMPERATURE,
SE_DT_INTERIOR_FLOW_VELOCITY,
SE_DT_LARGE_SCALE_CORRELATION,
SE_DT_LARGE_SCALE_ROUGHNESS,
SE_DT_LIGHT_LEVEL,
SE_DT_MAXIMUM_TEMPERATURE,
SE_DT_POLARIZATION_LABEL,
SE_DT_RADAR_CROSS_SECTION,
SE_DT_RADAR_SIGNIFCANT_FACTOR_LABEL,
SE_DT_RADIANCE,
SE_DT_RADIANCE_AMPLITUDE,
SE_DT_RADIANCE_AZIMUTH,
SE_DT_RADIANCE_DIFFUSED_LUNAR,
SE_DT_RADIANCE_DIFFUSED_SOLAR,
SE_DT_RADIANCE_DIRECT_LUNAR,
SE_DT_RADIANCE_DIRECT_SOLAR,
SE_DT_RADIANCE_ELEVATION,
SE_DT_RADIANCE_PHASE,
SE_DT_REAL_REFRACTIVE_INDEX,
SE_DT_REFLECTED_AZIMUTH,
SE_DT_REFLECTED_ELEVATION,
SE_DT_REFLECTIVITY,
SE_DT_RELATIVE_RADIANCE,
SE_DT_SECONDARY_TEXTURE,

SE_DT_SOLAR_ABSORPTIVITY,
SE_DT_SPECIFIC_HEAT,
SE_DT_SPECULAR_REFLECTIVITY,
SE_DT_SUN_SHADING,
SE_DT_SUPPORT_TEMP_CODE,
SE_DT_SURFACE_BACKSCATER,
SE_DT_THERMAL_CONDUCTIVITY,
SE_DT_THERMAL_CONTRAST,
SE_DT_THICKNESS,
SE_DT_TOTAL_TRANSMISIVITY,
SE_DT_TRANSMISIVITY,
SE_DT_TRANSMISSION_ATTENUATION,
SE_DT_TRANSMISSION_LOSS,
SE_DT_TRANSMITTED_AZIMUTH,
SE_DT_TRANSMITTED_ELEVATION,
SE_DT_VISUAL_CONTRAST,
SE_DT_VOLUME_BACKSCATTER,
SE_DT_WAVELENGTH

New SE_DATA_TABLE_VALUE_TYPE_ENUM entries:
SE_DT_POLARIZATION
SE_DT_RADAR_SIGNIFCANT_FACTOR
SE_DT_EM_BAND

New struct SE_DATA_TABLE_VALUE.u entries:
SE_DT_POLARIZATION_ENUM,
SE_DT_RADAR_SIGNIFCANT_FACTOR_ENUM
SE_DT_EM_BAND_ENUM

New struct SE_DATA_TABLE_VALUE_PTR.u entries:
SE_DT_POLARIZATION_ENUM,
SE_DT_RADAR_SIGNIFCANT_FACTOR_ENUM
SE_DT_EM_BAND_ENUM

Disposition (approve, disapprove, ...)
Rationale:

Disposition Date:

Disposition Authority:

SEDRIS CHIEF SOFTWARE ENGINEER Date

Appendix G

Sensor-Related SEDRIS Attribute Codes

SAC	Name	Label	Description	Unit_Code	Data_Type	Max	Min	Digits	Tol	Rationale
AADL	Angle, Azimuth - Radiance (Local)	SE_SAC_ANGLE_AZIMUTH_RADIANCE_LOCAL_NORMAL	The angle in the horizontal plane measured clockwise from the projection of the local surface normal to the projection of the radiance vector.	DEG_ARC	FLOAT32	359.99	0.00	5	0.01	Added to support EM/IR in Release 1.34; was "Radiance Azimuth Angle".
AADN	Angle, Azimuth - Radiance (True North)	SE_SAC_ANGLE_AZIMUTH_RADIANCE_TRUE_NORTH	The angle in the horizontal plane, measured clockwise from true north, to the projection of the radiance vector in the horizontal plane.	DEG_ARC	FLOAT32	359.99	0.00	5	0.01	Added to support EM/IR in Release 1.34; was "Radiance Azimuth Angle".
AAIL	Angle, Azimuth - Incident (Local)	SE_SAC_ANGLE_AZIMUTH_INCIDENT_LOCAL_NORMAL	The angle in the horizontal plane measured clockwise from the projection of the local surface normal to the projection of the incident vector.	DEG_ARC	FLOAT32	359.99	0.00	5	0.01	Added to support EM/IR in Release 1.34; was "Incident Azimuth Angle".
AAIN	Angle, Azimuth - Incident (True North)	SE_SAC_ANGLE_AZIMUTH_INCIDENT_TRUE_NORTH	The angle in the horizontal plane, measured clockwise from true north, to the projection of the incident vector in the horizontal plane.	DEG_ARC	FLOAT32	359.99	0.00	5	0.01	Added to support EM/IR in Release 1.34; was "Incident Azimuth Angle".
AARL	Angle, Azimuth - Reflected (Local)	SE_SAC_ANGLE_AZIMUTH_REFLECTED_LOCAL_NORMAL	The angle in the horizontal plane measured clockwise from the projection of the local surface normal to the projection of the reflected vector.	DEG_ARC	FLOAT32	359.99	0.00	5	0.01	Added to support EM/IR in Release 1.34; was "Reflected Azimuth Angle".
AARN	Angle, Azimuth - Reflected (True North)	SE_SAC_ANGLE_AZIMUTH_REFLECTED_TRUE_NORTH	The angle in the horizontal plane, measured clockwise from true north, to the projection of the reflected vector in the horizontal plane.	DEG_ARC	FLOAT32	359.99	0.00	5	0.01	Added to support EM/IR in Release 1.34; was "Reflected Azimuth Angle".
AATL	Angle, Azimuth - Transmitted (Local)	SE_SAC_ANGLE_AZIMUTH_TRANSMITTED_LOCAL_NORMAL	The angle in the horizontal plane measured clockwise from the projection of the local surface normal to the projection of the transmitted vector.	DEG_ARC	FLOAT32	359.99	0.00	5	0.01	Added to support EM/IR in Release 1.34; was "Transmitted Azimuth Angle".
AATN	Angle, Azimuth - Transmitted (True North)	SE_SAC_ANGLE_AZIMUTH_TRANSMITTED_TRUE_NORTH	The angle in the horizontal plane, measured clockwise from true north, to the projection of the transmitted vector in the horizontal plane.	DEG_ARC	FLOAT32	359.99	0.00	5	0.01	Added to support EM/IR in Release 1.34; was "Transmitted Azimuth Angle".
ADCM	Air Density, Climatology - Mean	SE_SAC_AIR_DENSITY_CLIMATOLOGY_MEAN	The mean ratio of the mass of air to the volume it occupies.	KG/M^3	FLOAT32	2.000000	0.000001	7	0	(Direct match to 44930) Added to integrate Data Tables with Properties for Release 1.34; was "Density Climatology Mean".
AEDH	Angle, Elevation - Radiance (Horizontal)	SE_SAC_ANGLE_ELEVATION_RADIANCE_HORIZONTAL	The angle in the vertical plane, measured from the horizontal plane, to the radiance vector.	DEG_ARC	FLOAT32	+90.0	0.0	3	0.1	Added to support EM/IR in Release 1.34; was "Radiance Elevation Angle".
AEDL	Angle, Elevation - Radiance (Local Normal)	SE_SAC_ANGLE_ELEVATION_RADIANCE_LOCAL_NORMAL	The angle in the vertical plane, measured from the local surface normal, to the radiance vector.	DEG_ARC	FLOAT32	+90.0	0.0	3	0.1	Added to support EM/IR in Release 1.34; was "Radiance Elevation Angle".

SAC	Name	Label	Description	Unit_Code	Data_Type	Max	Min	Digits	Tol	Rationale
AEIH	Angle, Elevation - Incident (Horizontal)	SE_SAC_ANGLE_ELEVATION_INCIDENT_HORIZONTAL	The angle in the vertical plane, measured from the horizontal plane, to the incident vector.	DEG_ARC	FLOAT32	+90.0	0.0	3	0.1	Added to support EM/IR in Release 1.34; was "Incident Elevation Angle".
AEIL	Angle, Elevation - Incident (Local)	SE_SAC_ANGLE_ELEVATION_INCIDENT_LOCAL_NORMAL	The angle in the vertical plane, measured from the local surface normal, to the incident vector.	DEG_ARC	FLOAT32	+90.0	0.0	3	0.1	Added to support EM/IR in Release 1.34; was "Incident Elevation Angle".
AERH	Angle, Elevation - Reflected (Horizontal)	SE_SAC_ANGLE_ELEVATION_REFLECTED_HORIZONTAL	The angle in the vertical plane, measured from the horizontal plane, to the reflected vector.	DEG_ARC	FLOAT32	+90.0	0.0	3	0.1	Added to support EM/IR in Release 1.34; was "Reflected Elevation Angle".
AERL	Angle, Elevation - Reflected (Local Normal)	SE_SAC_ANGLE_ELEVATION_REFLECTED_LOCAL_NORMAL	The angle in the vertical plane, measured from the local surface normal, to the reflected vector.	DEG_ARC	FLOAT32	+90.0	0.0	3	0.1	Added to support EM/IR in Release 1.34; was "Reflected Elevation Angle".
AETH	Angle, Elevation - Transmitted (Horizontal)	SE_SAC_ANGLE_ELEVATION_TRANSMITTED_HORIZONTAL	The angle in the vertical plane, measured from the horizontal plane, to the transmitted vector.	DEG_ARC	FLOAT32	+90.0	0.0	3	0.1	Added to support EM/IR in Release 1.34; was "Transmitted Elevation Angle".
AETL	Angle, Elevation - Transmitted (Local Normal)	SE_SAC_ANGLE_ELEVATION_TRANSMITTED_LOCAL_NORMAL	The angle in the vertical plane, measured from the local surface normal, to the transmitted vector.	DEG_ARC	FLOAT32	+90.0	0.0	3	0.1	Added to support EM/IR in Release 1.34; was "Transmitted Elevation Angle".
BDRF	Bidirectional Distribution Reflectance Function	SE_SAC_BIDIRECTIONAL_DISTRIBUTION_REFLECTANCE_FUNCTION	The ratio of reflected radiance to the irradiance.	1/SR	FLOAT32	100.000	0.000	6	0	Added to support EM/IR in Release 1.34; was "Bidirectional Reflectance Distribution Function".
BSU_	Backscatter, Surface	SE_SAC_BACKSCATTER_SURFACE	The ratio of the flux returned (reflected directly back in the direction of incidence) to the incident flux (in decibels, negative sign assumed).	DB	FLOAT32	400.0	0.0	4	0.1	Added to support EM/IR in Release 1.34; was "Surface Backscatter".
BVD_	Backscatter, Volume	SE_SAC_BACKSCATTER_VOLUME	The ratio of the flux, per meter of range, returned (reflected directly back in the direction of incidence) to the incident flux (in decibels, negative sign assumed). Due to interaction between the propagating flux and the transmitting medium.	DB/M	FLOAT32	400.0	0.0	4	0.01	Added to support EM/IR in Release 1.34; was "Volume Backscatter". Changed units from "dB/M" to "dB/Km" for consistency with definition.
CCO_	Convection Coefficient	SE_SAC_CONVECTION_COEFFICIENT	The ratio of convective heat flux density through a surface boundary to the temperature gradient across that boundary.	W/(M*K)	FLOAT32	1.00	0.00	3	0.01	Added to support EM/IR in Release 1.34; was "Convection Coefficient".
CTH_	Contrast, Temperature	SE_SAC_CONTRAST_TEMPERATURE	The contrast of an object expressed as the difference between the temperature of the object and the temperature of the apparent background.	K	FLOAT32	4000.00	0.00	6		Added to support EM/IR in Release 1.34; was "Thermal Contrast".

SAC	Name	Label	Description	Unit_Code	Data_Type	Max	Min	Digits	Tol	Rationale
CVI_	Contrast, Visual	SE_SAC_CONTRAST_VISUAL	The apparent contrast of an object expressed as a number between minus one (totally dark target) to zero (no apparent contrast) to infinity (very bright target).	UNITLESS	FLOAT32	+max(float32)	-1.0			Added to support EM/IR in Release 1.34; was "Visual Contrast".
DIR_	Directivity	SE_SAC_DIRECTIVITY	The side or sides of a feature which produces the greatest reflectivity potential to electromagnetic radiation.	ENUMERATION	ENUM					Added "to electromagnetic radiation" for clarification (e.g., radar or lidar).
EMB_	Electromagnetic Band (Code)	SE_SAC_ELECTROMAGNETIC_BAND	Identifies one band within a parititioning of the electromagnetic spectrum into standard frequency bands.	ENUMERATION	ENUM					Added to support EM/IR in Release 1.34; was "Electromagnetic Band Label".
EMS_	Emissivity	SE_SAC_EMISSIVITY	Ratio of emission of a sample to that of a blackbody.	UNITLESS	FLOAT32	1.000	0.000	4	0	Added to support EM/IR in Release 1.34; was "Emissivity".
EPT_	Electromagnetic Polarization Type	SE_SAC_ELECTROMAGNETIC_POLARIZATION_TYPE	Indicates the type of polarization of electromagnetic radiation. Polarization is the condition of having, e.g., uniform and non-random elliptical, circular, or linear variation in vibrational orientation of light or other electromagnetic radiation.	ENUMERATION	ENUM					(Related to 45370: 'SATELLITE-ELECTROMAGNETIC-SENSOR-TYPE-CHANNEL polarization code') Added to support EM/IR in Release 1.34; was "Polarization Label".
IFV_	Interior Flow Velocity	SE_SAC_INTERIOR_FLOW_VELOCITY	Speed of any internal convective flow within an object.	M/S	FLOAT32	20.000	0.000	3		Added to support EM/IR in Release 1.34; was "Interior Flow Velocity".
RCG_	Radar Clutter, Ground	SE_SAC_RADAR_CLUTTER_GROUND	The radar ground clutter expressed in average Radar Cross Section (RCS) per unit area of land.	UNITLESS	FLOAT32	1.0000	0.0000	5		Added to support EM/IR in Release 1.34; was "Ground Clutter".
RCO_	Radar Clutter, Ocean	SE_SAC_RADAR_CLUTTER_OCEAN	The radar ground clutter expressed in average Radar Cross Section (RCS) per unit area of ocean.	UNITLESS	FLOAT32	1.0000	0.0000	5		Added to support EM/IR in Release 1.34; was "Sea Clutter".
RCS_	Radar Cross Section	SE_SAC_RADAR_CROSS_SECTION	Power (or flux) reflected from an object toward the sensor per solid angle, divided by the incident power density times four Pi.	M^2	FLOAT32	1000.00	0.00	6	0.01	Added to support EM/IR in Release 1.34; was "Radar Cross Section".
RDD_	Radiation, Direct Downwelling	SE_SAC_RADIATION_DIRECT_DOWNWELLING	The irradiance from the sky onto an object.	W/M^2	FLOAT32	9.99	10^-6	3		Added to support EM/IR in Release 1.34; was "Direct Downwelling".
RDN_	Radiance	SE_SAC_RADIANCE	The radiant power per unit source area per unit solid angle.	W/(M^2*SR)	FLOAT32	10^15				Added to support EM/IR in Release 1.34; was "Radiance".
RFA_	Radiance, Fluctuation Amplitude	SE_SAC_RADIANCE_FLUCTUATION_AMPLITUDE	Value between 0 and 255 indicative of the amount of fluctuation (peak amplitude) of the surface radiance (or luminance) over a 24 hour period.	UNITLESS	UINT8	255	0	3	1	Added to support EM/IR in Release 1.34; was "Radiance Fluctuation Amplitude".

SAC	Name	Label	Description	Unit_Code	Data_Type	Max	Min	Digits	Tol	Rationale
RFS_	Roughness (Surface), Fine Scale	SE_SAC_ROUGHNESS_FINE_SCALE	The fine scale (i.e., within one or two orders of magnitude of the wavelength of the electromagnetic radiation of concern) roughness expressed as the standard deviation of the surface variations.	M	FLOAT32	10^3	10^-8			Added to support EM/IR in Release 1.34; was "Fine Scale Roughness".
RFSC	Roughness (Surface), Fine Scale - Correlation	SE_SAC_ROUGHNESS_FINE_SCALE_CORRELATION	The correlation length of the Fine Scale (surface) Roughness.	M	FLOAT32	10^3	10^-8			Added to support EM/IR in Release 1.34; was "Fine Scale Roughness Correlation".
RII_	Refractive Index, Imaginary Part	SE_SAC_REFRACTIVE_INDEX_IMAGINARY_PART	The imaginary part of the refractivity index. The refractivity index is the ratio of the wavelength or phase velocity of an electromagnetic wave in a vacuum to that in the medium or substance.	UNITLESS	FLOAT32	1.000	0.000	4	0	Added to support EM/IR in Release 1.34; was "Imaginary Part of Refractive Index".
RIR_	Refractive Index, Real Part	SE_SAC_REFRACTIVE_INDEX_REAL_PART	The real part of the refractivity index is the ratio of the wavelength or phase velocity of an electromagnetic wave in a vacuum to that in the medium or substance.	UNITLESS	FLOAT32	1.000	0.000	4	0	Added to support EM/IR in Release 1.34; was "Real Part of Refractive Index".
RLD_	Radiation, Lunar - Direct	SE_SAC_RADIATION_LUNAR_DIRECT	Direct lunar incident flux per unit of object surface area.	W/M^2	FLOAT32	9.99	0.00	3	0.01	Added to support EM/IR in Release 1.34; was "Direct Lunar Irradiance"
RLF_	Radiation, Lunar - Diffused	SE_SAC_RADIATION_LUNAR_DIFFUSED	Diffused lunar incident flux per unit of object surface area.	W/M^2	FLOAT32	9.99	0.00	3	0.01	Added to support EM/IR in Release 1.34; was "Diffused Lunar Irradiance"
RLS_	Roughness (Surface), Large Scale	SE_SAC_ROUGHNESS_LARGE_SCALE	The large scale (i.e., orders of magnitude larger than the wavelength of the electromagnetic radiation of concern) roughness expressed as the standard deviation of the surface variations.	M	FLOAT32	10^3	10^-8			Added to support EM/IR in Release 1.34; was "Large Scale Roughness".
RLSC	Roughness (Surface), Large Scale - Correlation	SE_SAC_ROUGHNESS_LARGE_SCALE_CORRELATION	The correlation length of the Large Scale (surface) Roughness.	M	FLOAT32	10^3	10^-8			Added to support EM/IR in Release 1.34; was "Large Scale Roughness Correlation".
RPH_	Radiance, Phase	SE_SAC_RADIANCE_PHASE	Value from 0 to 24 indicating phase (in 15 degree increments) of the radiance amplitude assuming a sine wave fluctuation.	HR	INT8	24	0	2	1	Added to support EM/IR in Release 1.34; was "Radiance Phase".
RSD_	Radiation, Solar - Direct	SE_SAC_RADIATION_SOLAR_DIRECT	Direct solar incident flux per unit of object surface area.	W/M^2	FLOAT32	9.99	0.00	3	0.01	Added to support EM/IR in Release 1.34; was "Direct Solar Irradiance"
RSF_	Radiation, Solar - Diffused	SE_SAC_RADIATION_SOLAR_DIFFUSED	Diffused solar incident flux per unit of object surface area.	W/M^2	FLOAT32	9.99	0.00	3	0.01	Added to support EM/IR in Release 1.34; was "Diffused Solar Irradiance".
RSI_	Radar Significance Factor	SE_SAC_RADAR_SIGNIFICANCE_FACTOR	Categorization of man-made or natural objects based on the object's predominant exposed surface.	ENUMERATION	ENUM					Added to support EM/IR in Release 1.34; was "Radar Significant Factor". Source: NIMA Digital Feature Analysis Data (DFAD) Specification

SAC	Name	Label	Description	Unit_Code	Data_Type	Max	Min	Digits	Tol	Rationale
RVB_	Radiation, Visible - Brightness	SE_SAC_RADIATION_VISIBLE_BRIGHTNESS	The quantity of visible light as processed by the normal human eye. Also known as light level, or illuminance. Illuminance is defined as radiation per unit time (flux) per unit area. Lux is defined as lumens per square meter.	LX	FLOAT32	10^6	10^-7	3		(Related to 45008: 'ILLUMINATION-ANALYSIS-FORECAST brightness quantity' and ???: 'SOLAR-LUNAR-ILLUMINANCE-PREDICTION illuminance quantity') Added to integrate Data Tables with Properties and support EM/IR for Release 1.34; was "Radiation Visible Light Brightness"
SAB_	Surface Absorptivity	SE_SAC_SURFACE_ABSORPTIVITY	The ratio of absorbed to incident flux at a surface (in decibels, negative sign assumed).	DB	FLOAT32	99.9	0.0	3	0.1	Added to support EM/IR in Release 1.34; was "Absorptivity".
SAS_	Surface Absorptivity - Solar	SE_SAC_SURFACE_ABSORPTIVITY_SOLAR	The ratio of absorbed to incident solar flux at a surface (in decibels, negative sign assumed).	DB	FLOAT32	99.9	0.0	3	0.1	Added to support EM/IR in Release 1.34; was "Solar Absorptivity".
SPH_	Specific Heat	SE_SAC_SPECIFIC_HEAT	The quantity of energy required to raise the temperature of a unit mass of the material one degree Kelvin.	J/(G*K)	FLOAT32					Added to support EM/IR in Release 1.34; was "Specific Heat".
SRE_	Surface Reflectivity	SE_SAC_SURFACE_REFLECTIVITY	The ratio of reflected to incident flux at a surface.	UNITLESS	FLOAT32	1.000	0.000	4	0	Added to support EM/IR in Release 1.34; was "Reflectivity".
SRF_	Surface Reflectivity, Diffused Component	SE_SAC_SURFACE_REFLECTIVITY_DIFFUSED_COMPONENT	The diffused component of the Surface Reflectivity.	UNITLESS	FLOAT32	1.000	0.000	4	0	Added to support EM/IR in Release 1.34; was "Diffused Component of Reflectivity".
SRS_	Surface Reflectivity, Specular	SE_SAC_SURFACE_REFLECTIVITY_SPECULAR	The specular component of the Surface Reflectivity.	UNITLESS	FLOAT32	1.000	0.000	4	0	Added to support EM/IR in Release 1.34; was "Specular Reflectivity".
STSS	Surface Thermal Sun Shading Scalar	SE_SAC_SURFACE_THERMAL_SUN_SHADING_SCALAR	The degree of dominance of the effect of sun shading (in the respective sensor domain) over the thermally derived values, ranged zero to one, with one being maximally (but not necessarily completely) dominant. The exact interpretation of the Surface Thermal Sun Shading Scalar value is determined by the Surface Thermal value Model (STVM) enumeration.	UNITLESS	FLOAT32	1.00	0.00	3	0.01	Added to support EM/IR in Release 1.34; was "Sun Shading Scalar".
STST	Surface Thermal Secondary Texture	SE_SAC_SURFACE_THERMAL_SECONDARY_TEXTURE	The degree of dominance of an image's visual texture (in the respective sensor domain) over the thermally derived values, ranged zero to one, with one being maximally (but not necessarily completely) dominant. The exact interpretation of the Surface Thermal Secondary Texture value is determined by the Surface Thermal value Model (STVM) enumeration.	UNITLESS	FLOAT32	1.00	0.00	3	0.01	Added to support EM/IR in Release 1.34; was "Secondary Texture".

SAC	Name	Label	Description	Unit_Code	Data_Type	Max	Min	Digits	Tol	Rationale
STV_	Surface Thermal Value	SE_SAC_SURFACE_THERMAL_VALUE	The index used to describe the thermal (radiance) charactistic of a surface. This value is used to index one or more thermal (radiance) model-specific lookup tables in order to predict, display, or analyze thermal signatures. The interpretation of the the thermal index is determined by the Surface Thermal value Model (STVM) enumeration.	UNITLESS	UINT8	255	0	3	1	Added to integrate Data Tables with Properties for Release 1.34; was "Thermal Value".
STVM	Surface Thermal Value Model	SE_SAC_SURFACE_THERMAL_VALUE_MODEL	Identifies the thermal (radiance) model for the associated Surface Thermal Value (STV_), thus determining the interpretation of the thermal index describing the thermal (radiance) charactistic of a surface. That value is used to index one or more thermal model-specific lookup tables in order to predict, display, pr analyze thermal signatures.	ENUMERATION	ENUM					Added to integrate Data Tables with Properties for Release 1.34; was "Thermal Value Description". Usage changed from an attempt to define the explicit semantic of each value, to defining the thermal model for which the value is applicable.
SUT_	Support Temperature Code	SE_SAC_SUPPORT_TEMPERATURE_CODE	An indicator of the primary physical mechanism accounting for the steady-state surface temperature.	ENUMERATION	ENUM					Added to support EM/IR in Release 1.34; was "Support Temperature Code"
TCO_	Thermal Conductivity	SE_SAC_THERMAL_CONDUCTIVITY	The ratio of thermal flux density through a surface, within a material or object, to the temperature gradient across that surface.	W/(M*K)	FLOAT32					Added to support EM/IR in Release 1.34; was "Thermal Conductivity".
TDD_	Thickness, Diurnal Depth	SE_SAC_THICKNESS_DIURNAL_DEPTH	Thermal thickness or effective depth of temperature variation within an object over the diurnal cycle; diurnal thermal fluctuations cease around three to five diural depths.	M	FLOAT32	10000.000	0.000	8	0	Added to support EM/IR in Release 1.34; was "Diurnal Depth".
TEI_	Temperature, Interior	SE_SAC_TEMPERATURE_INTERIOR	The temperature of the interior of a material or object.	K	FLOAT32	5000.0	-273.15			Added to support EM/IR in Release 1.34; was "Interior Temperature".
TEM_	Temperature, Maximum	SE_SAC_TEMPERATURE_MAXIMUM	The maximum temperature a particular material or object can achieve.	K	FLOAT32	5000.0	-273.15			Added to support EM/IR in Release 1.34; was "Maximum Temperature".
TES_	Temperature, Surface	SE_SAC_TEMPERATURE_SURFACE	The temperature of the surface of a material or object.	K	FLOAT32	5000.0	-273.15			Added to support EM/IR in Release 1.34 (for consistency with other additions).
THI_	Thickness	SE_SAC_THICKNESS	The dimension between two of an object's opposite surfaces; usually the dimension of smallest measure.	M	FLOAT32	10^6	10^-12	5		Added to support EM/IR in Release 1.34; was "Thickness".
TRN_	Transmissivity	SE_SAC_TRANSMISSIVITY	The ratio of transmitted flux to incident flux, per meter of material or object thickness.	1/M	FLOAT32	1.000	0.000	4	0	Added to support EM/IR in Release 1.34; was "Transmissivity".

SAC	Name	Label	Description	Unit_Code	Data_Type	Max	Min	Digits	Tol	Rationale
TRNA	Transmission Attenuation	SE_SAC_TRANSMISSION_ATTENUATION	The ratio of transmitted flux to incident flux, in decibels per meter of material or object thickness.	DB/M	FLOAT32	200.0	0.0	4	0.1	Added to support EM/IR in Release 1.34; was "Transmission Attenuation".
TRNL	Transmission Loss	SE_SAC_TRANSMISSION_LOSS	The total energy loss due to absorption, scattering, or other phenomena, associated with transmission through a material or object.	DB	FLOAT32	200.0	0.0	4	0.1	Added to support EM/IR in Release 1.34; was "Transmission Loss".
TRNT	Total Transmissivity	SE_SAC_TRANSMISSIVITY_TOTAL	The ratio of transmitted flux to incident flux from one traversal through a material or object.	UNITLESS	FLOAT32	1.000	0.000	4	0.01	Added to support EM/IR in Release 1.34; was "Total Transmissivity".
WAV_	Wavelength	SE_SAC_WAVELENGTH	The distance in a periodic wave between two points of corresponding phase in consecutive cycles.	M	FLOAT32					Added to support EM/IR in Release 1.34; was "Wavelength".

Appendix H

Minimal PTN Database Computer Code and Data Values

Minimal PTN Database Computer Code:

```
#include <stdio.h>
#include <math.h>
#include "simple_ptn.h"

int read_ptn(file, vertex_list, nvertices)
char *file;
PTN_VERTEX **vertex_list;
long *nvertices;
{
 long j;
 int eof_flag=0;

 PTN_VERTEX vbuffer;
 FILE *fptr;

 fptr = fopen(file,"r");
 if (fptr==NULL) {printf("error opening PTN file ...\n");
exit(-1);}

(*vertex_list) = (PTN_VERTEX *)malloc(sizeof(PTN_VERTEX)*1);
 if (*vertex_list == NULL)
 {
   printf("error allocating initial VERTEX_LIST array\n");
   exit(-1);
 }
 j = 0;
 while(eof_flag!=EOF)
  {
   eof_flag = fscanf(fptr,"%f",&vbuffer.x);
   eof_flag = fscanf(fptr,"%f",&vbuffer.y);
   eof_flag = fscanf(fptr,"%f",&vbuffer.z);
   eof_flag = fscanf(fptr,"%f",&vbuffer.i);
 eof_flag = fscanf(fptr,"%f",&vbuffer.j);
 eof_flag = fscanf(fptr,"%f",&vbuffer.k);
 eof_flag = fscanf(fptr,"%f",&vbuffer.eL);
 eof_flag = fscanf(fptr,"%f",&vbuffer.elwir);
 eof_flag = fscanf(fptr,"%f",&vbuffer.emwir);
 eof_flag = fscanf(fptr,"%f",&vbuffer.aS);
 eof_flag = fscanf(fptr,"%f",&vbuffer.h);
 eof_flag = fscanf(fptr,"%f",&vbuffer.kt);
 eof_flag = fscanf(fptr,"%f",&vbuffer.p);
 eof_flag = fscanf(fptr,"%f",&vbuffer.ch);
```

```
if (eof_flag!=EOF)
{
j++;
(*vertex_list) =(PTN_VERTEX*)realloc((*vertex_list),sizeof(PTN_VERTEX)*j);
(*vertex_list)[j-1].x = vbuffer.x;
(*vertex_list)[j-1].y = vbuffer.y;
(*vertex_list)[j-1].z = vbuffer.z;
(*vertex_list)[j-1].i = vbuffer.i;
(*vertex_list)[j-1].j = vbuffer.j;
(*vertex_list)[j-1].k = vbuffer.k;
(*vertex_list)[j-1].eL = vbuffer.eL;
(*vertex_list)[j-1].elwir = vbuffer.elwir;
(*vertex_list)[j-1].emwir = vbuffer.emwir;
(*vertex_list)[j-1].aS = vbuffer.aS;
(*vertex_list)[j-1].h = vbuffer.h;
(*vertex_list)[j-1].kt = vbuffer.kt;
(*vertex_list)[j-1].p = vbuffer.p;
(*vertex_list)[j-1].ch = vbuffer.ch;
/***********
 printf("Vertex %ld :\n", j);
 printf("%lf\n", vbuffer.x );
 printf("%lf\n", vbuffer.y );
 printf("%lf\n", vbuffer.z );
 printf("%lf\n", vbuffer.i );
 printf("%lf\n", vbuffer.j );
 printf("%lf\n", vbuffer.k );
 printf("%lf\n", vbuffer.eL );
 printf("%lf\n", vbuffer.elwir );
 printf("%lf\n", vbuffer.emwir );
 printf("%lf\n", vbuffer.aS );
 printf("%lf\n", vbuffer.h );
 printf("%lf\n", vbuffer.kt );
 printf("%lf\n", vbuffer.p );
 printf("%lf\n", vbuffer.ch );
*****************/
}
}

*nvertices = j;
fclose(fptr);
return(-1);
}
```

```
int main(int argc, char *argv[])
{
 char ptn_file[100];
 long i,n;
 PTN_VERTEX *vertex_list;

 if (argc<2)
  {printf("%s <PTN simple file> \n", argv[0]);
   exit(-1);
  }

 strcpy(ptn_file, argv[1]);
 read_ptn(ptn_file, &vertex_list, &n);

 /***************************
 for (i=0; i<n; i++)
 {
 printf("Vertex %ld :\n", i);
 printf("%lf\n", vertex_list[i].x );
 printf("%lf\n", vertex_list[i].y );
 printf("%lf\n", vertex_list[i].z );
 printf("%lf\n", vertex_list[i].i );
 printf("%lf\n", vertex_list[i].j );
 printf("%lf\n", vertex_list[i].k );
 printf("%lf\n", vertex_list[i].eL );
 printf("%lf\n", vertex_list[i].elwir );
 printf("%lf\n", vertex_list[i].emwir );
 printf("%lf\n", vertex_list[i].aS );
 printf("%lf\n", vertex_list[i].h );
 printf("%lf\n", vertex_list[i].kt );
 printf("%lf\n", vertex_list[i].p );
 printf("%lf\n", vertex_list[i].ch );
 }
 ***************************/
 for (i=0; i<n; i++)
  printf("Vertex %ld :\t", i);
 printf("\n");
 for (i=0; i<n; i++)
  printf("%lf\t", vertex_list[i].x );
 printf("\n");

 for (i=0; i<n; i++)
  printf("%lf\t", vertex_list[i].y );
 printf("\n");
```

```
for (i=0; i<n; i++)
 printf("%lf\t", vertex_list[i].z );
printf("\n");

for (i=0; i<n; i++)
 printf("%lf\t", vertex_list[i].i );
printf("\n");

for (i=0; i<n; i++)
 printf("%lf\t", vertex_list[i].j );
printf("\n");

for (i=0; i<n; i++)
 printf("%lf\t", vertex_list[i].k );
printf("\n");

for (i=0; i<n; i++)
 printf("%lf\t", vertex_list[i].eL );
printf("\n");

for (i=0; i<n; i++)
 printf("%lf\t", vertex_list[i].elwir);
printf("\n");

for (i=0; i<n; i++)
 printf("%lf\t", vertex_list[i].emwir );
printf("\n");

for (i=0; i<n; i++)

 printf("%lf\t", vertex_list[i].aS );
printf("\n");

for (i=0; i<n; i++)
 printf("%lf\t", vertex_list[i].h );
printf("\n");

for (i=0; i<n; i++)
 printf("%lf\t", vertex_list[i].k );
printf("\n");

for (i=0; i<n; i++)
 printf("%lf\t", vertex_list[i].p );
```

```
printf("\n");

for (i=0; i<n; i++)
 printf("%lf\t", vertex_list[i].ch );printf("\n");

}
```

Minimal PTN Database Header File:

```
#include <stdio.h>

typedef struct{
float x, y, z; /* position in UTM and elevation meters */
float i, j, k; /* normal vector */
float eL; /* 3-oo um emissivity */
float elwir; /* 8-12 um emissivity */
float emwir; /* 3-5 um emissivity */
float aS; /* encoded absorptivity */
float h; /* coefficient of convection */
float kt; /* thermal conductivity */
float p; /* density */
float ch; /* specific heat */
}PTN_VERTEX;
```

Minimal PTN Database Data Values:

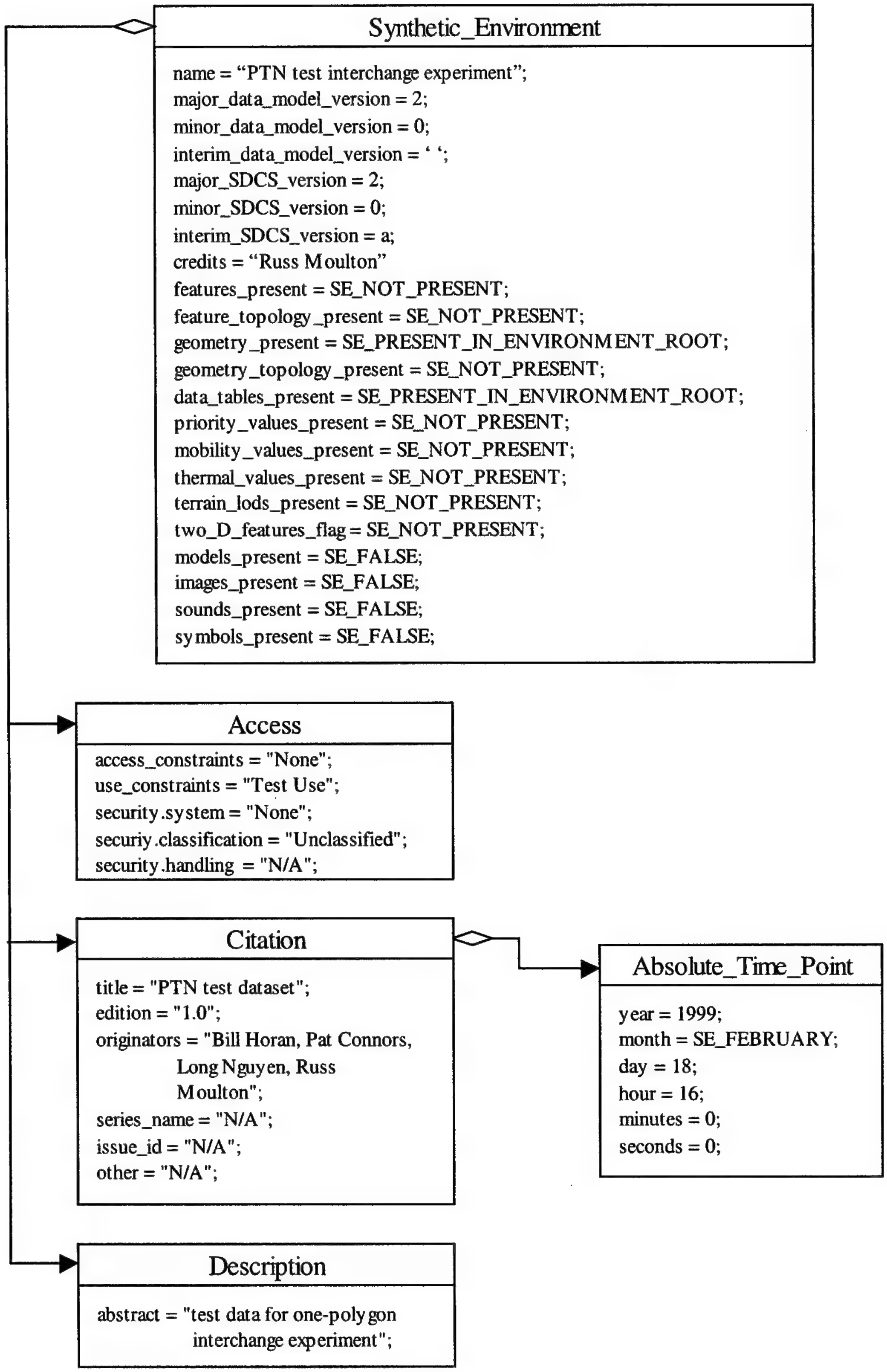
```
0
0
0
0
0
1
.92
.8
.5
.2
4
.52
1840
1500
```

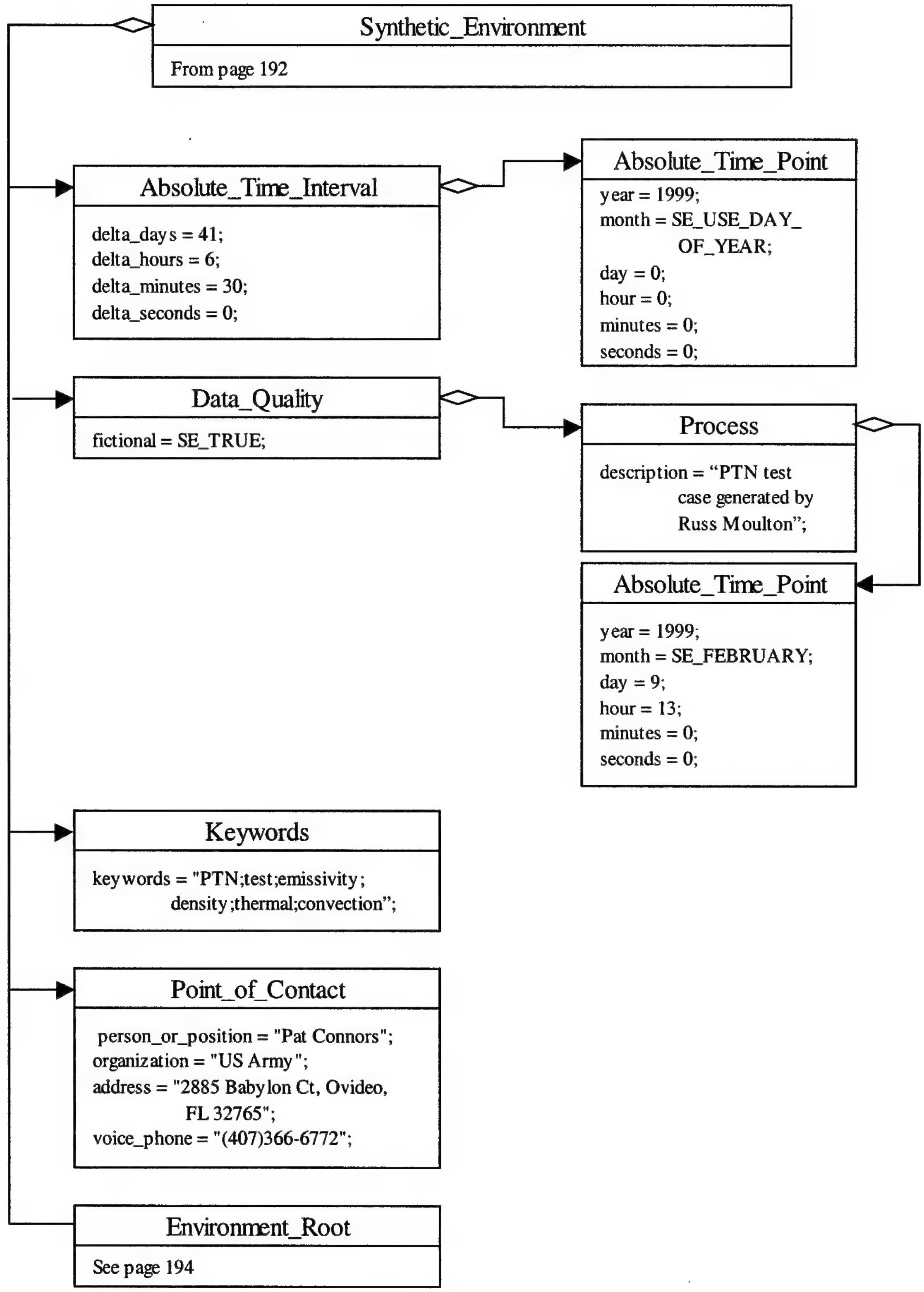
1
0
0
.577
-.577
.577
.85
.8
.5
.2
4
.062
920
1104

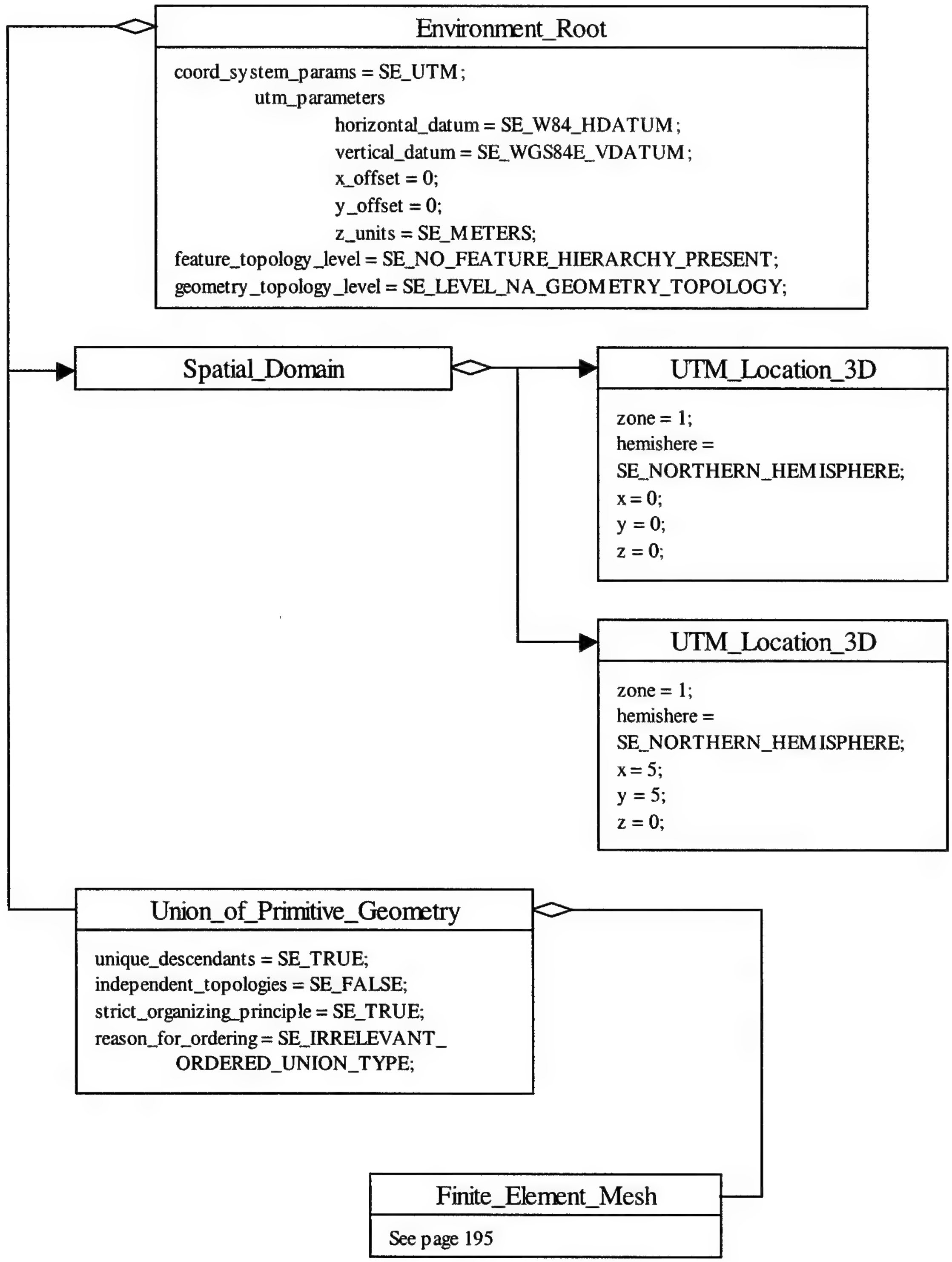
0
1
0
-.577
.577
.577
.8
.8
.5
.2
4
.062
920
1104

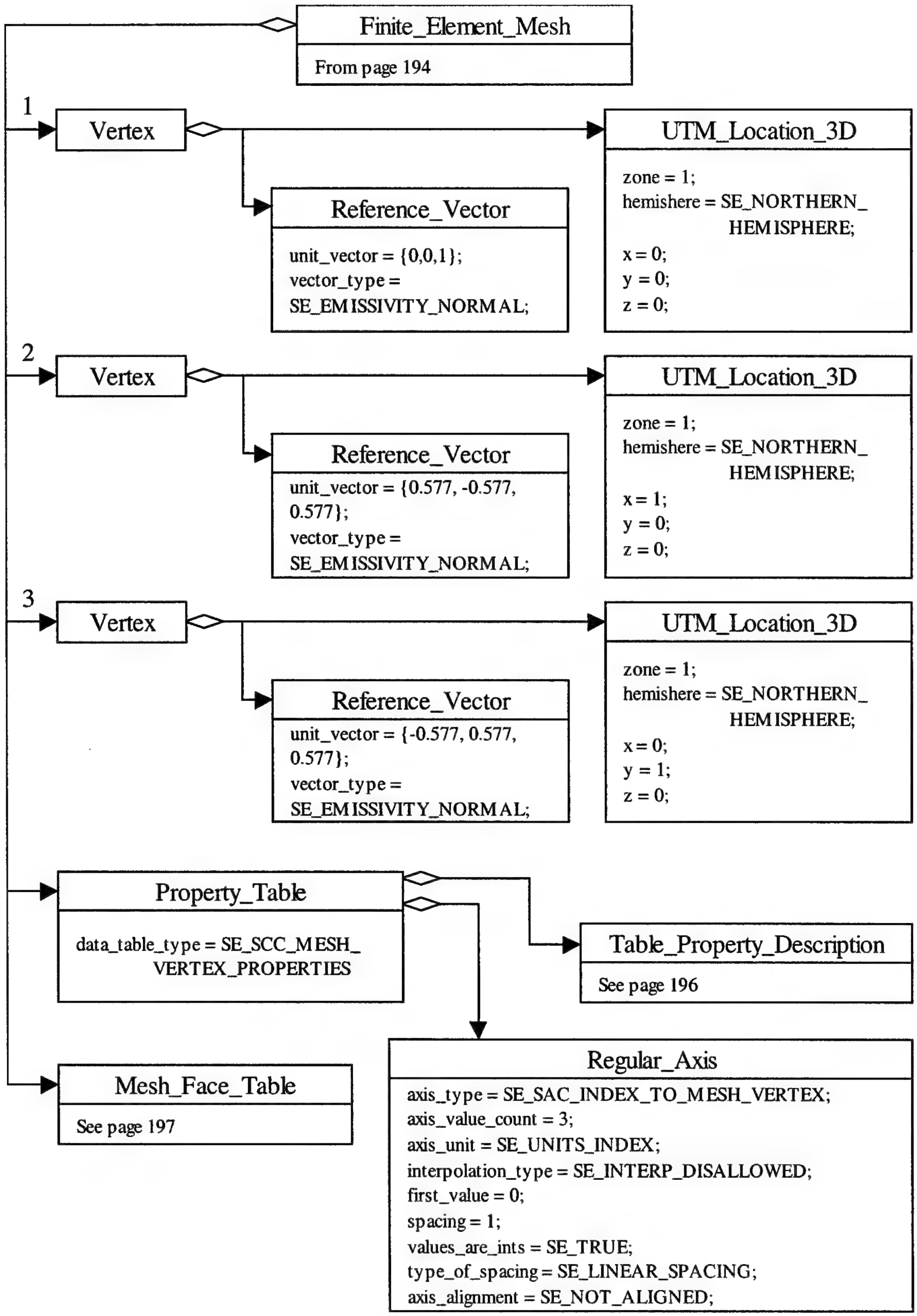
Appendix I

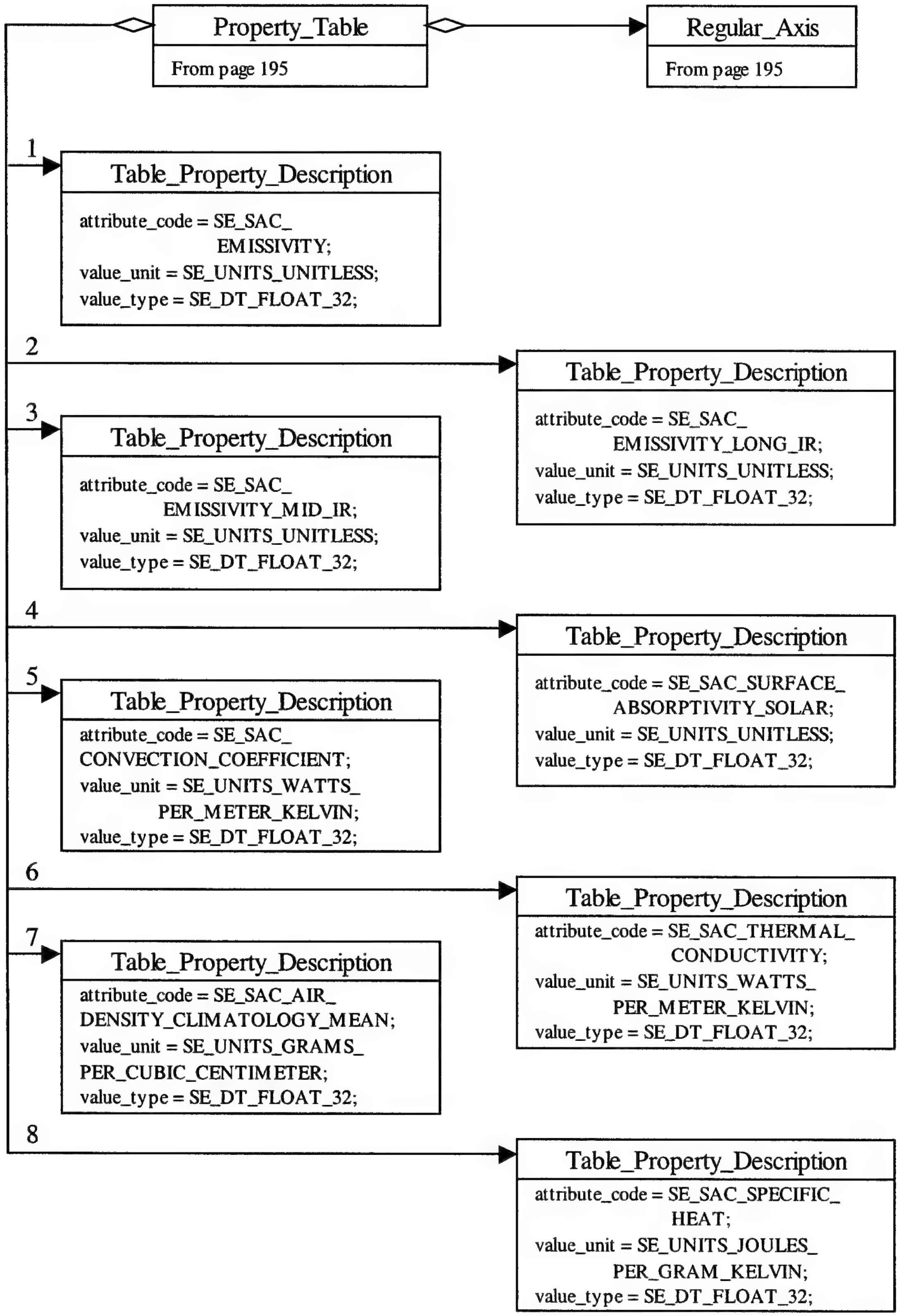
PTN Minimal Database Interchange Experiment Object Diagram

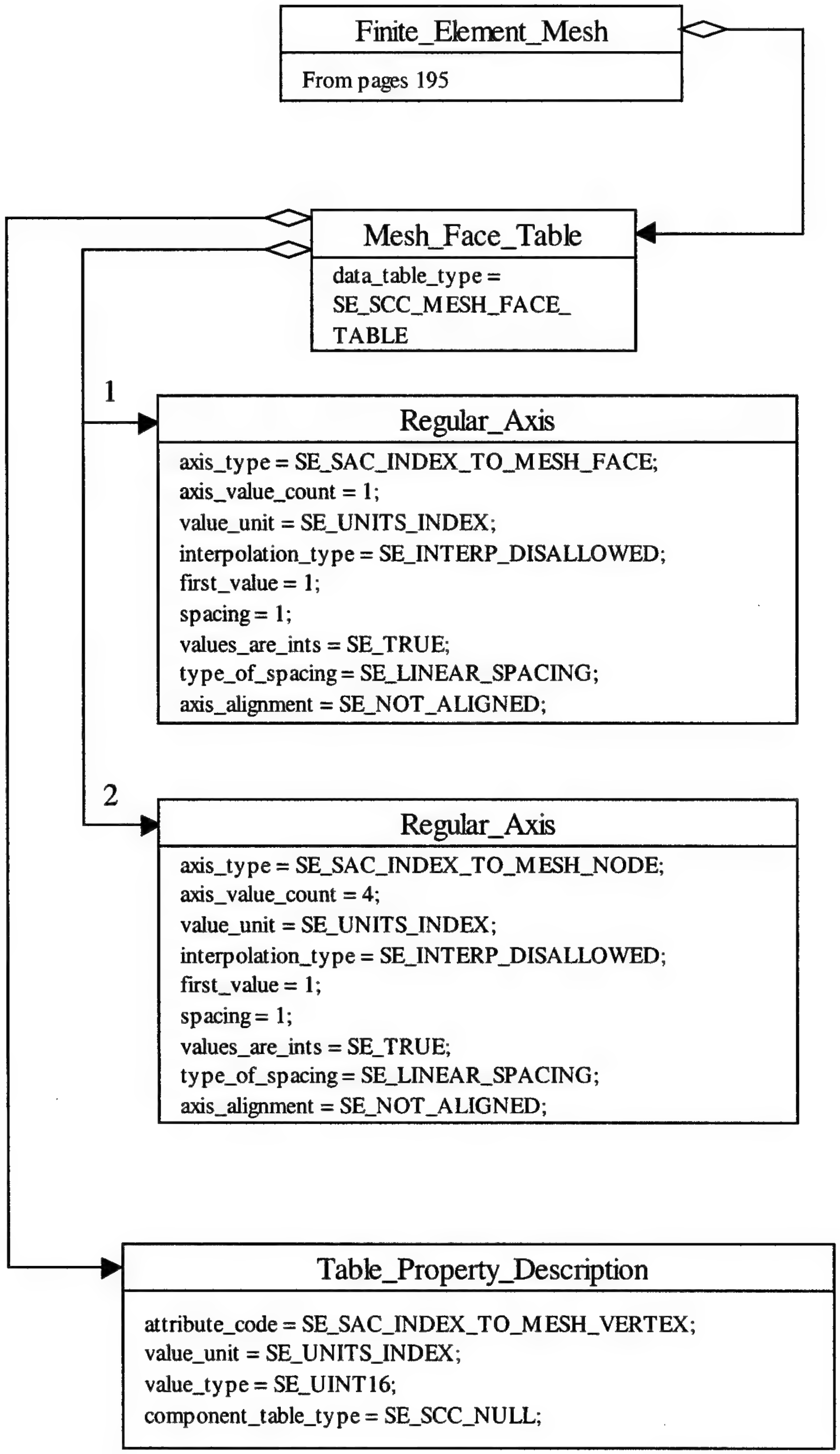

Synthetic_Environment
name = “PTN test interchange experiment”;
major_data_model_version = 2;
minor_data_model_version = 0;
interim_data_model_version = ‘ ‘;
major_SDCS_version = 2;
minor_SDCS_version = 0;
interim_SDCS_version = a;
credits = “Russ Moulton”
features_present = SE_NOT_PRESENT;
feature_topology_present = SE_NOT_PRESENT;
geometry_present = SE_PRESENT_IN_ENVIRONMENT_ROOT;
geometry_topology_present = SE_NOT_PRESENT;
data_tables_present = SE_PRESENT_IN_ENVIRONMENT_ROOT;
priority_values_present = SE_NOT_PRESENT;
mobility_values_present = SE_NOT_PRESENT;
thermal_values_present = SE_NOT_PRESENT;
terrain_lods_present = SE_NOT_PRESENT;
two_D_features_flag = SE_NOT_PRESENT;
models_present = SE_FALSE;
images_present = SE_FALSE;
sounds_present = SE_FALSE;
symbols_present = SE_FALSE;
Access
access_constraints = "None";
use_constraints = "Test Use";
security.system = "None";
securiy.classification = "Unclassified";
security.handling = "N/A";
Citation
title = "PTN test dataset";
edition = "1.0";
originators = "Bill Horan, Pat Connors,
Long Nguyen, Russ
Moulton";
series_name = "N/A";
issue_id = "N/A";
other = "N/A";
Absolute_Time_Point
year = 1999;
month = SE_FEBRUARY;
day = 18;
hour = 16;
minutes = 0;
seconds = 0;
Description
abstract = "test data for one-polygon
interchange experiment";


Synthetic_Environment
From page 192
Absolute_Time_Interval
delta_days = 41;
delta_hours = 6;
delta_minutes = 30;
delta_seconds = 0;
Absolute_Time_Point
year = 1999;
month = SE_USE_DAY_
OF_YEAR;
day = 0;
hour = 0;
minutes = 0;
seconds = 0;
Data_Quality
fictional = SE_TRUE;
Process
description = “PTN test
case generated by
Russ Moulton”;
Absolute_Time_Point
year = 1999;
month = SE_FEBRUARY;
day = 9;
hour = 13;
minutes = 0;
seconds = 0;
Keywords
keywords = "PTN;test;emissivity;
density;thermal;convection”;
Point_of_Contact
person_or_position = "Pat Connors";
organization = "US Army";
address = "2885 Babylon Ct, Ovideo,
FL 32765";
voice_phone = "(407)366-6772";
Environment_Root
See page 194


Environment_Root
coord_system_params = SE_UTM;
utm_parameters
horizontal_datum = SE_W84_HDATUM;
vertical_datum = SE_WGS84E_VDATUM;
x_offset = 0;
y_offset = 0;
z_units = SE_METERS;
feature_topology_level = SE_NO_FEATURE_HIERARCHY_PRESENT;
geometry_topology_level = SE_LEVEL_NA_GEOMETRY_TOPOLOGY;
Spatial_Domain
UTM_Location_3D
zone = 1;
hemishere =
SE_NORTHERN_HEMISPHERE;
x = 0;
y = 0;
z = 0;
UTM_Location_3D
zone = 1;
hemishere =
SE_NORTHERN_HEMISPHERE;
x = 5;
y = 5;
z = 0;
Union_of_Primitive_Geometry
unique_descendants = SE_TRUE;
independent_topologies = SE_FALSE;
strict_organizing_principle = SE_TRUE;
reason_for_ordering = SE_IRRELEVANT_
ORDERED_UNION_TYPE;
Finite_Element_Mesh
See page 195


Finite_Element_Mesh
From page 194
1
Vertex
UTM_Location_3D
zone = 1;
hemishere = SE_NORTHERN_HEMISPHERE;
x = 0;
y = 0;
z = 0;
Reference_Vector
unit_vector = {0,0,1};
vector_type = SE_EMISSIVITY_NORMAL;
2
Vertex
UTM_Location_3D
zone = 1;
hemishere = SE_NORTHERN_HEMISPHERE;
x = 1;
y = 0;
z = 0;
Reference_Vector
unit_vector = {0.577, -0.577, 0.577};
vector_type = SE_EMISSIVITY_NORMAL;
3
Vertex
UTM_Location_3D
zone = 1;
hemishere = SE_NORTHERN_HEMISPHERE;
x = 0;
y = 1;
z = 0;
Reference_Vector
unit_vector = {-0.577, 0.577, 0.577};
vector_type = SE_EMISSIVITY_NORMAL;
Property_Table
data_table_type = SE_SCC_MESH_VERTEX_PROPERTIES
Table_Property_Description
See page 196
Regular_Axis
axis_type = SE_SAC_INDEX_TO_MESH_VERTEX;
axis_value_count = 3;
axis_unit = SE_UNITS_INDEX;
interpolation_type = SE_INTERP_DISALLOWED;
first_value = 0;
spacing = 1;
values_are_ints = SE_TRUE;
type_of_spacing = SE_LINEAR_SPACING;
axis_alignment = SE_NOT_ALIGNED;
Mesh_Face_Table
See page 197


Property_Table
From page 195
Regular_Axis
From page 195
1
Table_Property_Description
attribute_code = SE_SAC_EMISSIVITY;
value_unit = SE_UNITS_UNITLESS;
value_type = SE_DT_FLOAT_32;
2
Table_Property_Description
attribute_code = SE_SAC_EMISSIVITY_LONG_IR;
value_unit = SE_UNITS_UNITLESS;
value_type = SE_DT_FLOAT_32;
3
Table_Property_Description
attribute_code = SE_SAC_EMISSIVITY_MID_IR;
value_unit = SE_UNITS_UNITLESS;
value_type = SE_DT_FLOAT_32;
4
Table_Property_Description
attribute_code = SE_SAC_SURFACE_ABSORPTIVITY_SOLAR;
value_unit = SE_UNITS_UNITLESS;
value_type = SE_DT_FLOAT_32;
5
Table_Property_Description
attribute_code = SE_SAC_CONVECTION_COEFFICIENT;
value_unit = SE_UNITS_WATTS_PER_METER_KELVIN;
value_type = SE_DT_FLOAT_32;
6
Table_Property_Description
attribute_code = SE_SAC_THERMAL_CONDUCTIVITY;
value_unit = SE_UNITS_WATTS_PER_METER_KELVIN;
value_type = SE_DT_FLOAT_32;
7
Table_Property_Description
attribute_code = SE_SAC_AIR_DENSITY_CLIMATOLOGY_MEAN;
value_unit = SE_UNITS_GRAMS_PER_CUBIC_CENTIMETER;
value_type = SE_DT_FLOAT_32;
8
Table_Property_Description
attribute_code = SE_SAC_SPECIFIC_HEAT;
value_unit = SE_UNITS_JOULES_PER_GRAM_KELVIN;
value_type = SE_DT_FLOAT_32;


Finite_Element_Mesh
From pages 195
Mesh_Face_Table
data_table_type =
SE_SCC_MESH_FACE_
TABLE
1
Regular_Axis
axis_type = SE_SAC_INDEX_TO_MESH_FACE;
axis_value_count = 1;
value_unit = SE_UNITS_INDEX;
interpolation_type = SE_INTERP_DISALLOWED;
first_value = 1;
spacing = 1;
values_are_ints = SE_TRUE;
type_of_spacing = SE_LINEAR_SPACING;
axis_alignment = SE_NOT_ALIGNED;
2
Regular_Axis
axis_type = SE_SAC_INDEX_TO_MESH_NODE;
axis_value_count = 4;
value_unit = SE_UNITS_INDEX;
interpolation_type = SE_INTERP_DISALLOWED;
first_value = 1;
spacing = 1;
values_are_ints = SE_TRUE;
type_of_spacing = SE_LINEAR_SPACING;
axis_alignment = SE_NOT_ALIGNED;
Table_Property_Description
attribute_code = SE_SAC_INDEX_TO_MESH_VERTEX;
value_unit = SE_UNITS_INDEX;
value_type = SE_UINT16;
component_table_type = SE_SCC_NULL;

Appendix J

PTN Test Interchange Experiment Transmittal Printout

```
global_directly_attach_table_components = False
global_process_inheritance = False
global_transform_locations = False
global_follow_model_instances = False
global_evaluate_static_control_links = False
Opened SEDRIS Transmittal PTN test interchange experiment.sens_db.
- Synthetic Environment
  name->string_length: 39
  name->string_value: PTN test interchange experiment.sens_db
  major_data_model_version: 2
  minor_data_model_version: 0
  interim_data_model_version:
  major_SDCS_version: 2
  minor_SDCS_version: 0
  interim_SDCS_version: a
  credits->string_length: 13
  credits->string_value: Russ Moulton
  features_present: Not Present
  feature_topology_present: Not Present
  geometry_present: Present In Environment Root
  geometry_topology_present: Not Present
  data_tables_present: Present In Environment Root
  priority_values_present: Not Present
  mobility_values_present: Not Present
  thermal_values_present: Not Present
  terrain_lods_present: Not Present
  two_D_features_flag: Not Present
  models_present: False
  images_present: False
  sounds_present: False
  symbols_present: False
  - Absolute Time Interval
    delta_days: 41
    delta_hours: 6
    delta_minutes: 30
    delta_seconds:      0.00000
    - Absolute Time Point
      year: 1999
      month: Use Day Of Year
      day: 0
      hour: 0
      minutes: 0
      seconds:      0.00000
  - Access
    access_constraints->string_length: 4
    access_constraints->string_value: None
    use_constraints->string_length: 8
    use_constraints->string_value: Test Use
    security->system->string_length: 4
    security->system->string_value: None
    security->classification->string_length: 12
    security->classification->string_value: Unclassified
    security->handling->string_length: 3
    security->handling->string_value: N/A
```

```
  - Citation
    title->string_length: 16
    title->string_value: PTN test dataset
    edition->string_length: 3
    edition->string_value: 1.0
    originators->string_length: 50
    originators->string_value: Bill Horan, Pat Connors, Long Nguyen,
Russ Moulton
    series_name->string_length: 3
    series_name->string_value: N/A
    issue_id->string_length: 3
    issue_id->string_value: N/A
    other->string_length: 3
    other->string_value: N/A
    - Absolute Time Point
      year: 1999
      month: February
      day: 18
      hour: 16
      minutes: 0
      seconds:      0.00000
  - Data Quality
    fictional: True
    field_accuracy->string_length: 0
    field_accuracy->string_value:
    logical_consistency->string_length: 0
    logical_consistency->string_value:
    completeness->string_length: 0
    completeness->string_value:
    abs_horiz_pos_accuracy->string_length: 0
    abs_horiz_pos_accuracy->string_value:
    rel_horiz_pos_accuracy->string_length: 0
    rel_horiz_pos_accuracy->string_value:
    abs_vert_pos_accuracy->string_length: 0
    abs_vert_pos_accuracy->string_value:
    rel_vert_pos_accuracy->string_length: 0
    rel_vert_pos_accuracy->string_value:
    - Process
      description->string_length: 39
      description->string_value: PTN test case generated by Russ
Moulton
      - Absolute Time Point
        year: 1999
        month: February
        day: 9
        hour: 13
        minutes: 0
        seconds:      0.00000
  - Description
    abstract->string_length: 48
    abstract->string_value: test data for one-polygon interchange
experiment
    purpose->string_length: 3
    purpose->string_value: N/A
```

```
    other->string_length: 3
    other->string_value: N/A
  - Keywords
    keywords->string_length: 46
    keywords->string_value:
PTN;test;emissivity;density;thermal;convection
  - Point of Contact
    person_or_position->string_length: 11
    person_or_position->string_value: Pat Connors
    organization->string_length: 7
    organization->string_value: US Army
    address->string_length: 34
    address->string_value: 2885 Babylon Ct., Oviedo, FL 32765
    voice_phone->string_length: 14
    voice_phone->string_value: (407) 366-6772
    fax_phone->string_length: 0
    fax_phone->string_value:
    tdd_tty_phone->string_length: 0
    tdd_tty_phone->string_value:
    email_address->string_length: 23
    email_address->string_value: pconnors@sprintmail.com
    web_site->string_length: 0
    web_site->string_value:
    hours_of_service->string_length: 0
    hours_of_service->string_value:
    other->string_length: 0
    other->string_value:
  - Environment Root
    coord_system_params.coord_system: UTM
    coord_system_params.u.utm_parameters.horizontal_datum: W84 HDatum
    coord_system_params.u.utm_parameters.vertical_datum: WGS84E VDatum
    coord_system_params.u.utm_parameters.x_offset:      0.00000
    coord_system_params.u.utm_parameters.y_offset:      0.00000
    coord_system_params.u.utm_parameters.z_units: Meters
    feature_topology_level: No Feature Hierarchy Present
    geometry_topology_level: Level NA Geometry Topology
    - Spatial Domain
      - UTM Location 3D
        zone: 1
        hemisphere: NORTHERN HEMISPHERE
        x:       0.00000
        y:       0.00000
        z:       0.00000
      - UTM Location 3D
        zone: 1
        hemisphere: NORTHERN HEMISPHERE
        x:       5.00000
        y:       5.00000
        z:       0.00000
    - Union of Primitive Geometry
      unique_descendants: True
      independent_topologies: False
      strict_organizing_principle: True
      reason_for_ordering: Irrelevant Ordered Union Type
```

```
- Finite Element Mesh
  - Vertex
    - UTM Location 3D
      zone: 1
      hemisphere: NORTHERN HEMISPHERE
      x:       0.00000
      y:       0.00000
      z:       0.00000
    - Reference Vector
      unit_vector->vector[0..3]:      0.00000       0.00000
1.00000
      vector_type: Emissivity Normal
  - Vertex
    - UTM Location 3D
      zone: 1
      hemisphere: NORTHERN HEMISPHERE
      x:       1.00000
      y:       0.00000
      z:       0.00000
    - Reference Vector
      unit_vector->vector[0..3]:      0.57700      -0.57700
0.57700
      vector_type: Emissivity Normal
  - Vertex
    - UTM Location 3D
      zone: 1
      hemisphere: NORTHERN HEMISPHERE
      x:       0.00000
      y:       1.00000
      z:       0.00000
    - Reference Vector
      unit_vector->vector[0..3]:     -0.57700       0.57700
0.57700
      vector_type: Emissivity Normal
  - Property Table
    data_table_type->tag[0..5]: XX999
    EMS_:       0.92000
    EMSA:       0.80000
    EMSB:       0.50000
    SAS_:       0.20000
    CCO_:       4.00000
    TCO_:       0.52000
    ADCM:    1840.00000
    SPH_:    1500.00000

    EMS_:       0.85000
    EMSA:       0.80000
    EMSB:       0.50000
    SAS_:       0.20000
    CCO_:       4.00000
    TCO_:       0.06200
    ADCM:     920.00000
    SPH_:    1104.00000
```

```
EMS_:       0.80000
EMSA:       0.80000
EMSB:       0.50000
SAS_:       0.20000
CCO_:       4.00000
TCO_:       0.06200
ADCM:     920.00000
SPH_:    1104.00000

- Regular Axis
  axis_type->tag[0..4]: INMV
  axis_unit: Units Index
  axis_value_count: 3
  interpolation_type: Interp Disallowed
  first_value:      0.00000
  spacing:      1.00000
  values_are_ints: True
  type_of_spacing: Linear Spacing
  axis_alignment: Not Aligned
- Table Property Description
  attribute_code->tag[0..4]: EMS_
  value_unit: Units unitless
  value_type: DT Float 32
  component_data_table_scc->tag[0..5]: _____
- Table Property Description
  attribute_code->tag[0..4]: EMSA
  value_unit: Units unitless
  value_type: DT Float 32
  component_data_table_scc->tag[0..5]: _____
- Table Property Description
  attribute_code->tag[0..4]: EMSB
  value_unit: Units unitless
  value_type: DT Float 32
  component_data_table_scc->tag[0..5]: _____
- Table Property Description
  attribute_code->tag[0..4]: SAS_
  value_unit: Units unitless
  value_type: DT Float 32
  component_data_table_scc->tag[0..5]: _____
- Table Property Description
  attribute_code->tag[0..4]: CCO_
  value_unit: Units Watts per meter Kelvin
  value_type: DT Float 32
  component_data_table_scc->tag[0..5]: _____
- Table Property Description
  attribute_code->tag[0..4]: TCO_
  value_unit: Units Watts per meter Kelvin
  value_type: DT Float 32
  component_data_table_scc->tag[0..5]: _____
- Table Property Description
  attribute_code->tag[0..4]: ADCM
  value_unit: Units grams per cubic centimeter
  value_type: DT Float 32
  component_data_table_scc->tag[0..5]: _____
```

```
    - Table Property Description
      attribute_code->tag[0..4]: SPH_
      value_unit: Units Joules per gram Kelvin
      value_type: DT Float 32
      component_data_table_scc->tag[0..5]: _____
  - Mesh Face Table
    data_table_type->tag[0..5]: XX999
     1

     2

     3

     1

    - Regular Axis
      axis_type->tag[0..4]: INMF
      axis_unit: Units Index
      axis_value_count: 1
      interpolation_type: Interp Disallowed
      first_value:      1.00000
      spacing:      1.00000
      values_are_ints: True
      type_of_spacing: Linear Spacing
      axis_alignment: Not Aligned
    - Regular Axis
      axis_type->tag[0..4]: INMN
      axis_unit: Units Index
      axis_value_count: 4
      interpolation_type: Interp Disallowed
      first_value:      1.00000
      spacing:      1.00000
      values_are_ints: True
      type_of_spacing: Linear Spacing
      axis_alignment: Not Aligned
    - Table Property Description
      attribute_code->tag[0..4]: INMV
      value_unit: Units Index
      value_type: DT U Int 16
      component_data_table_scc->tag[0..5]: _____

object      class
count       name
---------------------------------------------------------------------
      1     Absolute Time Interval
      3     Absolute Time Point
      1     Access
      1     Citation
      1     Data Quality
      1     Description
      1     Environment Root
      1     Finite Element Mesh
      1     Keywords
      1     Mesh Face Table
```

1	Point of Contact
1	Process
1	Property Table
3	Reference Vector
3	Regular Axis
1	Spatial Domain
1	Synthetic Environment
9	Table Property Description
1	Union of Primitive Geometry
5	UTM Location 3D
3	Vertex

41

Maximum Tree Height = 6

Appendix K

PTN Target Database Output

```
SEDRIS 132% ptn PTN_test_interchange_experiment.stf
Max Number of Active Buffers: 2048
Max Number of Active Files: 50
FileManager::RegisterFile - filename [PTN_test_interchange_experiment.stf]

*** This File is using STF_BIG_ENDIAN ***
*** This Machine is STF_BIG_ENDIAN ***

Root has no baseFileName.
Number of files 1
Initialized ObjectMap size to 98317
 0.000000
 0.000000
 0.000000
 0.000000
 0.000000
 1.000000
 0.920000
 0.800000
 0.500000
 0.200000
 4.000000
 0.520000
 1840.000000
 1500.000000

 1.000000
 0.000000
 0.000000
 0.577000
 -0.577000
 0.577000
 0.850000
 0.800000
 0.500000
 0.200000
 4.000000
 0.062000
 920.000000
 1104.000000

 0.000000
 1.000000
 0.000000
```

-0.577000
0.577000
0.577000
0.800000
0.800000
0.500000
0.200000
4.000000
0.062000
920.000000
1104.000000

LIST OF REFERENCES

Ada Information Clearinghouse (AdaIC). (1996). *Ada Used to Develop Visual and Sensor Displays* (Flyer U158-0596). Falls Church, VA: Ada Joint Program Office.

Allen, J. A., Hays, R.T., & Buffardi, L.C. (1986). Maintenance training simulator fidelity and individual differences in transfer of training. *Human Factors, 28* (5), 497-509.

Birkel, P. A. (1998, March). *The Synthetic Environment: Why We Are Here ....* Presentation given at the SEDRIS Open House, Orlando, FL.

Birkel, P. A. (1999, February). *SEDRIS Data Coding Standard: Classifications, and Features, Attributes and Values, and States. Using the Data Representation Model to describe specific things....* Presentation given at the SEDRIS Technology Conference, Orlando, FL.

Bhatti, S. N. (1994). "The electromagnetic spectrum; propagation in free-space and the atmosphere; noise in free-space." Unpublished lecture notes, Department of Computer Science, University College, London.

Cornette, W. M. (1996). *Modeling & Simulation Scene Generation Attributes Standard,* (Preliminary Draft). Bethesda, MD: Terrain Modeling Project Office, National Imagery & Mapping Agency.

Crum, S. (1997). "Energy Sources and Radiometric Principles, Remote Sensing Core Curriculum, Volume 2: Overview of Remote Sensing of the Environment." Unpublished lecture notes, Department of Geography, University of Texas at Austin.

Department of Defense. (1995). *Modeling and Simulation (M & S) Master Plan,* (DoD 5000.59-P). Washington, DC: Under Secretary of Defense for Acquisition and Technology.

Digital Geographic Information Working Group. (1997). *The Digital Geographic Information Exchange Standard (DIGEST)* (Part 4 of Edition 2.0, June 1997). Directorate of Geomatics, Department of National Defence, Canada.

Dynetics, Inc. (1991). *Weather Effects on EO/IR/MMW Sensors* [Brochure]. Huntsville, AL: Author.

Ellis, J. R. (1994). *Objectifying Real-Time Systems* (R. S. Weiner, Ed.). New York: SIGS.

Foley, P. G., Mamaghani, F., & Birkel, P. A. (1998, April). *The Synthetic Environment Data Representation and Interchange Specification (SEDRIS) Development Project.* Paper presented at the Ninth Annual International Training and Education Conference (ITEC), Lausanne, Switzerland.

Frieden, D. R. (1985). *Principles of Naval Weapons Systems.* Annapolis, MD: Naval Institute Press.

Guralnik, D. B. (Ed.). (1970). *Webster's New World Dictionary of the American Language* (2nd ed.). Englewood Cliffs, NJ: Prentice-Hall.

Holst, G. C. (1995). *Electro-Optical Imaging System Performance.* Winter Park, FL: JCD Publishing.

James, R. H. (1997a). *SEDRIS Data Model.* Unpublished paper, Orion Development Group, Inc.

James, R. H. (1997b). *The Synthetic Environment.* Unpublished paper, Orion Development Group, Inc.

James, R. H. (1997c). *Trainer Architecture Evolution.* Unpublished paper, Orion Development Group, Inc.

Mamaghani, F. (1994). What makes virtual systems a reality. *Computer Graphics, 28* (2), 105-109.

Mamaghani, F. (1998a). Creation and use of synthetic environments in realtime networked interactive simulation. In C. Dodsworth (Ed.), *Digital Illusion: Entertaining the Future with High Technology* (pp. 99-113). New York, NY: ACM Press.

Mamaghani, F. (1998b, March). *Overview of SEDRIS: Why is SEDRIS Needed?.* Presentation given at the SEDRIS Open House, Orlando, FL.

Piplani, L. K., Mercer, J. G., Roop, R. O. (1994). *Systems Acquisition Manager's Guide for the use of Models and Simulations* (Report of the DSMC 1993-1994 Military Research Fellows). Fort Belvoir, VA: Defense Systems Management College Press.

Rogatto, W. D. (Ed.). (1993). *The Infrared & Electro-Optical Systems Handbook: Vol. 3. Electro-Optical Components*. Copub. Ann Arbor, MI: Environmental Research Institute of Michigan. Bellingham, WA: SPIE Optical Engineering Press.

SEDRIS Data Coding Standard (Version 2.0) [Electronic data file]. (1999). Orlando, FL: The SEDRIS Development Team. Birkel, P. A. The MITRE Corporation. [Producer and Distributor].

SEDRIS Data Dictionary (Version 1.04d) [Electronic data file]. (1997). Orlando, FL: The SEDRIS Development Team. Horan, W. H., & Worley, M. L. [Producer and Distributor].

SEDRIS Data Dictionary (Version 2.0) [Electronic data file]. (1999). Orlando, FL: The SEDRIS Development Team. Horan, W. H., & Worley, M. L. [Producer and Distributor].

SEDRIS Data Model (Version 1.04d) [Electronic data file]. (1997). Orlando, FL: The SEDRIS Development Team. Horan, W. H., & Worley, M. L. [Producer and Distributor].

SEDRIS Data Model (Version 2.0) [Electronic data file]. (1999). Orlando, FL: The SEDRIS Development Team. Horan, W. H., & Worley, M. L. [Producer and Distributor].

SEDRIS technical information web site. (1998). *SEDRIS Homepage* [On-line]. Available: http://www.sedris.org/

Terrain Modeling Project Office (TMPO). (1996). *Modeling and Simulation Terrain Execution Plan* (Version 96-4.1). Washington, DC: Defense Mapping Agency.

U.S. Army Simulation, Training, and Instrumentation Command [STRICOM]. (1998a). *SEDRIS and The Synthetic Environment Domain* (Volume 1 of the SEDRIS Documentation Set, 28 March 1998). Orlando, FL: Author.

U.S. Army Simulation, Training, and Instrumentation Command. (1998b). *Synthetic Environment Data Representation and Interchange Specification Overview* (Volume 2 of the SEDRIS Documentation Set, 28 March 1998). Orlando, FL: Author.

U.S. Army Simulation, Training, and Instrumentation Command. (1998c). *SEDRIS Common Glossary* (Volume 5 of the SEDRIS Documentation Set, 28 March 1998). Orlando, FL: Author.

Welch, M. R. (1997a). *SEDRIS as an Interchange Mechanism.* Unpublished paper, Orion Development Group, Inc.

Welch, M. R. (1997b). *Synthetic Environment Databases*. Unpublished paper, Orion Development Group, Inc.

Welch, M. R., & James, R. H. (1997). *SEDRIS Vision*. Unpublished paper, Orion Development Group, Inc.